OCTOBER 1984 VOL. 2, NO. 4 $3.95

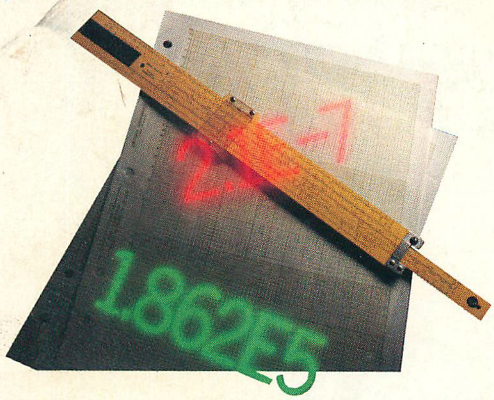
FOR IBM PERSONAL COMPUTER USERS

PC TECH JOURNAL

[redacted]

SIGNIFICANT FIGURES, I

Floating-point addition and subtraction on the IBM PC



RELAY: POWERFUL COMMUNICATIONS

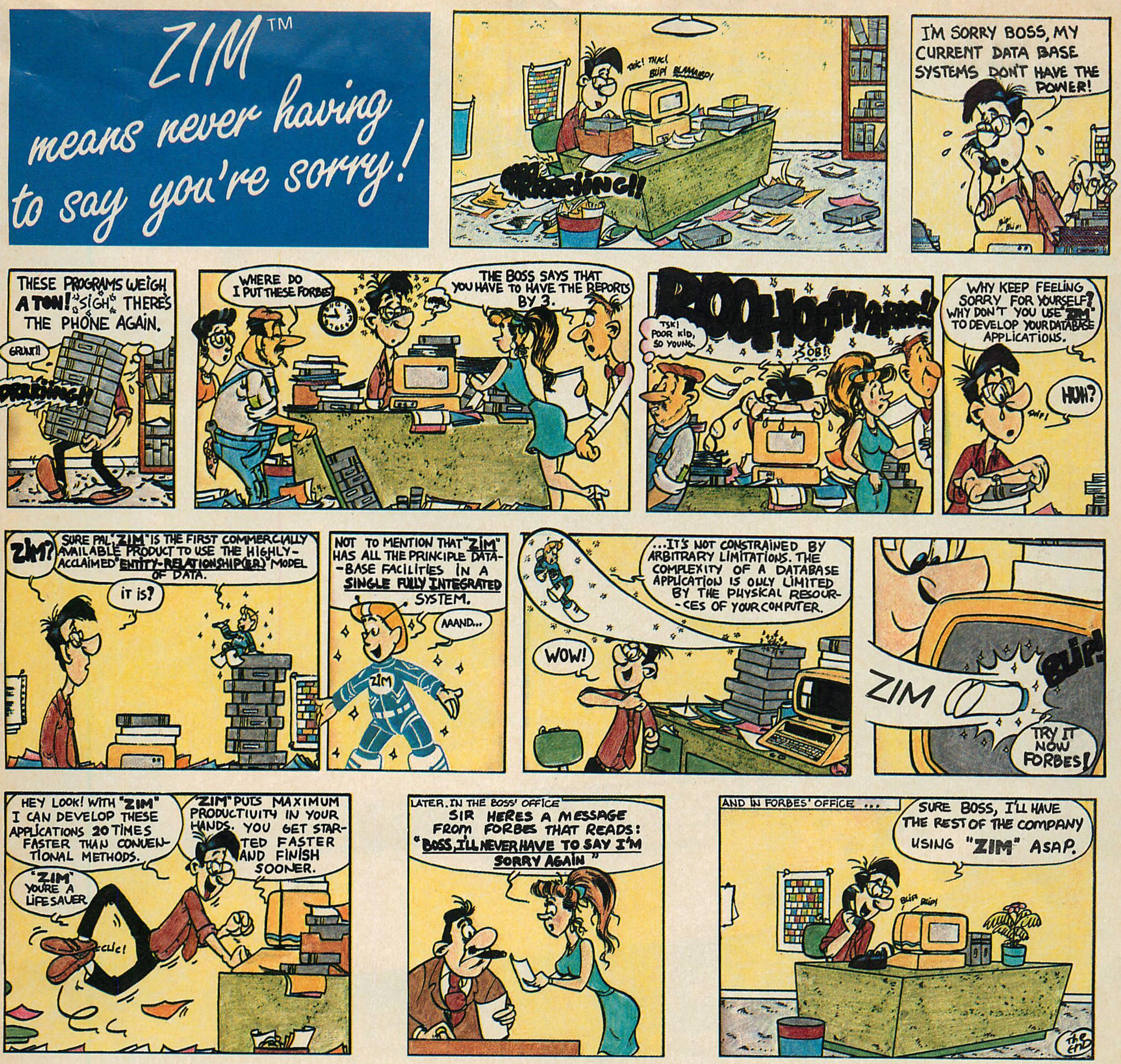
TWO OBJECT-MODULE LIBRARIANS

THE TRANSITION FROM BASIC TO C

IBM PC PASCAL FORMATTERS

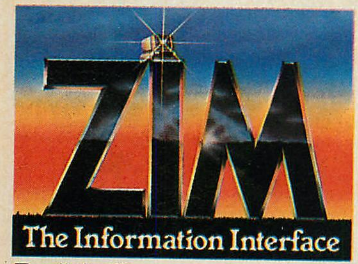
MULTITASKING WITH MULTILINK

From start to finish
The winner is . . .



Simply put, **ZIM** is the most modern, comprehensive and efficient 4th Generation Database Applications Development Language available.

ZIM's Entity-Relationship model gives you the most powerful database design and data manipulation capabilities with unlimited capacity for growth.



You get exceptional ease of use because **ZIM** integrates the functions of query, update, forms painting and manipulation, reporting and data dictionary into a uniform and elegant English-like command language.

ZIM puts maximum productivity in your hands. You get started faster and finish sooner. And of course, you will be backed by top notch support from Zanthe — the company that understands what "real" database application development is all about.

You'll never have to say "sorry" to yourself again.

ZANTHE
INFORMATION INC.
1785 Woodward Drive
Ottawa, Ontario K2C 0R1 (613) 727-1397

CIRCLE NO. 148 ON READER SERVICE CARD

ZIM is available on most personal computers under the MS-DOS operating system. A multi-user version of **ZIM** is available for the QNX and UNIX multi-user operating systems

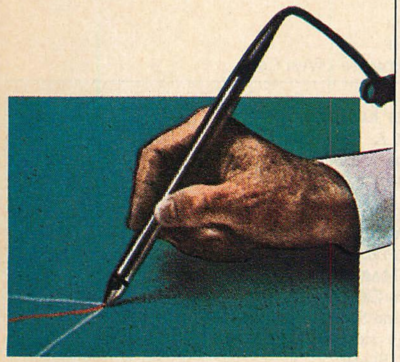
FOR IBM PERSONAL COMPUTER USERS

Volume 2, Number 4
October 1984

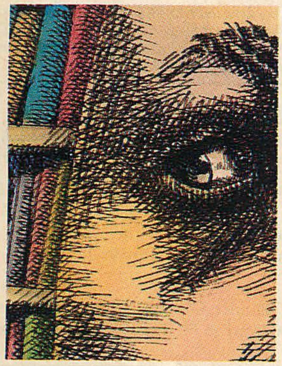
TECH JOURNAL

ARTICLES

22



36



80

DEPARTMENTS

TECH NOTEBOOKS

PRODUCTS

PC TECH JOURNAL

VOL. 2, NO. 4

Publisher: NEWTON BARRETT

Editorial
Editor: WILL FASTIE
Managing Editor: MARJORY SPRAYCAR
Technical Editors: JULIE ANDERSON, SUSAN GLINERT-COLE
Senior Copy Editor: SUSAN HOLLY
Copy Editor: BARBARA TILLY
Research Assistant: ARTHUR A. GLECKLER
Contributing Editors: RAY DUNCAN, RICHARD FOARD, AUGIE HANSEN, THOMAS V. HOFFMANN, SOL LIBES, WILLIAM H. MURRAY, MAX STUL OPPENHEIMER,
Editorial Secretary: DIANA L. CAREY
Editorial Assistant: CAROLE AUTENZIO

Art & Production
Art Director: NANCY LEPOW
Assistant Art Director: JANE FREY
Art Assistant: SANDRA RAY

Advertising Sales
Assistant to Publisher: JULIE HENDERSON
Advertising Coordinator: SHERYL PANZER
District Managers:
RITA BURKE, IAN SMITH—East Coast
CAROLINE ANDERSON—Midwest
TED BAHR, HARRIET ROGERS—West Coast
Account Sales Manager: PAULINE SCHERER
Account Representatives:
JAN SCHULTZ—East Coast
JOHN GROGAN—Midwest
BILL BUSH, ARLENE STEADMAN—West Coast

Circulation
Subscription Director: CHET KLIMUSZKO
Direct Mail Manager: ERIC A. BERNHARD
Renewal and Billing Manager: SHANE BOEL
Media Manager: ANN PICHETTE
Special Subscription Sales: DANIEL ROSENSWEIG

Computer Publications Division
General Manager: KEN KOPPEL
Vice President: EILEEN G. MARKOWITZ
Vice President, Circulation: CAROLE MANDEL
Vice President, Licensing and Special Projects: JERRY SCHNEIDER
Vice President, Creative Services: HERBERT STERN
Creative Director: PETER J. BLANK
Editorial Director: JONATHAN D. LAZARUS
Marketing Manager: RONNI SONNENBERG

Ziff-Davis Publishing
President, RICHARD P. FRIESE; President, Consumer Magazine Division, ALBERT S. TRAINA; Executive Vice President, Marketing and Circulation, PAUL H. CHOOK; Senior Vice President, PHILLIP T. HEFFERNAN; Senior Vice President, SIDNEY HOLTZ; Senior Vice President, EDWARD D. MUHLFELD; Senior Vice President, PHILIP SINE; Vice President, BAIRD DAVIS; Vice President, GEORGE MORRISSEY; Vice President, RORY PARISI; Vice President, WILLIAM L. PHILLIPS; Treasurer, SELWYN TAUBMAN; Secretary, BERTRAM A. ABRAMS

Editorial Office
PC TECH JOURNAL, The World Trade Center, Suite 211, Baltimore, MD 21202. 301-576-0770. The Source ID STY682. CompuServe 74156, 2365.

Advertising Offices
(East Coast) *PC TECH JOURNAL*, One Park Avenue, New York, NY 10016. 212-503-5100.
(Midwest) 180 N. Michigan Avenue, Chicago, IL 60601. 312-346-2600.
(West Coast) 11 Davis Drive, Belmont, CA 94002. 415-594-2290.

Subscription Inquiries
PC TECH JOURNAL, P.O. Box 2968, Boulder, CO 80321. Subscription service: 212-503-5300. Back issues: send $7.00/copy to *PC TECH JOURNAL*, Box CN, 1914, Morristown, NJ 07960.

PC TECH JOURNAL (ISSN 0738-0194) is published 12 times a year, $29.97 for one year, $52.97 for two years, $69.97 for three years. Additional postage $12 for Canada & Foreign by Ziff-Davis Publishing Company, One Park Avenue, New York, NY 10016. Application for Second-Class Postage Rates pending at New York, NY 10001. POSTMASTER: Send address changes or subscription inquiries to *PC TECH JOURNAL*, P.O. Box 2968, Boulder, CO 80321.

BPA

Graphics

An interview with the editor of a leading computer magazine

The extent to which graphics on the IBM PC are big business seems to be the extent to which spreadsheets with graphics capabilities are used. Although a number of excellent programs exist for two- and three-dimensional drafting, business charting, and drawing, the graphics capabilities of the PC inhibit such products from taking the market by storm. No one I talk to thinks of the PC as a graphics engine, citing limited processor power and insufficient screen resolution as the basis for their conclusion.

I've been mulling over this fact for some time. I assumed that visiting the big trade shows for computers (NCC, Comdex) would provide some answers—that I would see how well the PC was accepted as a system for graphics applications. In point of fact, I saw very little.

That's when I hit upon the idea of visiting the exposition at SIGGRAPH '84 in Minneapolis. I reasoned that SIGGRAPH, which is perhaps the most prestigious graphics conference in the world, would provide me a perspective on the state of the art in graphics and an opportunity to prowl the show floor looking for PCs.

Now that I'm back, you would think that I would have some conclusions for you. Not so, I'm afraid. That left me with a quandary: without conclusions, how could I draft a piece of prose for this space?

I couldn't. But while looking through my Reporter's Notebook at all the notes I had scribbled as I traveled those miles and miles of booths, I realized that there was another way out. I may not have the answers, but boy, do I know the right questions.

The bad news is that, outside of the cabbie who drove me to the airport in Minneapolis—or my seat partner, who slept the entire way home—I had no one to interview but myself. "Would you, Mr. Fastie, agree to an interview for an upcoming issue of *PC Tech Journal*?" I asked myself.

Why not? First question, please.

PCTJ: What has been the history of graphics on the IBM PC?

Fastie: One of limited processor power and insufficient graphics resolution, I'm afraid.

PCTJ: I see. Let me rephrase the question. In your original review of the IBM PC, published in Creative Computing, *you praised the graphics capabilities of the PC and compared them favorably with the other machines then available. Why is your attitude different now?*

Fastie: We've just passed the third birthday of the IBM PC. A lot has happened since the machine first came out, most particularly with regard to the cost of memory chips. Because resolution and color are both dependent upon the amount of memory that can be dedicated to a video controller, better capability usually means more memory. Three years ago, that was a problem. Today, it is not.

Furthermore the capabilities of the PC upon its birth were better in some ways but worse in others than prevailing standards for small computers. For example, the resolution was better but the flexibility in the use of color was not as good.

The problem is that IBM has not improved the situation with the PC line. PC*jr* shows some improvement, but other compromises in its design reduce the power of PC*jr* as a graphics engine.

PCTJ: So does that mean that the IBM PC is unsuitable as a graphics engine?

Fastie: Not quite, and that's the main answer I learned at SIGGRAPH. What I discovered was a rash of PCs being used in many graphics systems for a variety of purposes. That indicates to me that the IBM PC has gained tremendous acceptance in the graphics community, not so much as a machine that can handle graphics, but as a standard frame upon which to build graphics systems. For example, most of the PCs I saw at SIGGRAPH sported a high-power graphics board, obviously not IBM's, and a third-party, high-resolution display.

Equally important, IBM PCs were the *only* small computers I saw being used in this fashion, with one exception (not including vendors' booths—such as DEC's, which was showing only DEC's own computers). The exception was one Apple Lisa. I was surprised to note that the Lisa, with its sophisticated

graphics capabilities, was not being used as a graphics engine.

PCTJ: I don't understand. Why should suppliers of graphics systems be willing to settle for an IBM PC if it is underpowered?

Fastie: I think they recognize that IBM is the most desirable computer from the buyer's point of view. So they jump on IBM's coattails and solve the power problem in their own way. By the way, these vendors had crowded booths at SIGGRAPH. Clearly, the IBM PCs sitting in their booths were a draw.

PCTJ: Did you spot any trends at SIGGRAPH '84? What seemed to be the hot items?

Fastie: I'm not sure that I detected any trends other than the use of PCs as a basic computer frame. As for hot items, there were three, one of which was, again, all the PCs I saw. The other . . .

PCTJ: Excuse me. The trade press, including your sister publication PC Week, *has reported the presence of PCs. Those reports indicated a few PCs, not many, as you seem to imply. How many did you actually see?*

Fastie: I'd say that about 20 or 30 booths had a PC. Considering what was reported to me about the previous SIGGRAPH, and adding to that what I have seen at other trade shows, I'd call that so-called "small" number explosive.

As for the other hot items, one is unrelated to the PC market, at least for the moment. Much emphasis was being placed on the integration of video with computer-generated graphics and animation. I think we are in for some continuing innovations in that area and, of course, the eventual use of PCs in that field, as well.

The other hotbed was graphics hard-copy devices. Prices are falling, and sophistication and quality are rising. There was even a color laser printer, something I had heard about but never seen before. These products will tend to increase the downward migration of graphics applications to small computers.

PCTJ: You said a moment ago that IBM had not improved the PC's graphics since it was announced. Is IBM standing still?

Fastie: Quite the contrary. IBM is moving into this area very quickly. However, the first item on its graphics agenda relates, as it always has in other areas, to the company's large system business.

IBM has been aggressive of late with announcements of graphics workstations, some even based on the PC. The most visible, to coin a phrase, is the 3270-PC/G. This is a machine designed to be attached to a larger IBM system. Sure, it's a PC, but it is surely not meant to be used in the same way as a stand-alone, desktop PC would be.

I should also point out that IBM has always offered sophisticated graphics terminals to its mainframe customers, so the technology of graphics is certainly not unknown to the company.

PCTJ: What do you think will happen to the PC?

Fastie: I wish I knew. I was hopeful that a new adapter and display would be announced by IBM for the PC, but the rumors I've been hearing seem to contradict that. Most of these rumors center around the nature of the display subsystem for the new model PC, the one supposedly based on the Intel 80286 chip. That is supposed to have a higher resolution and good color.

PCTJ: What should IBM do with the new PC or, for that matter, with the current machine?

Fastie: I think it would be in IBM's best interest to develop a better, higher resolution standard for the PC, one that can be carried successfully into the new model as well. I think the minimum adequate specifications are 350 dots vertically by 720 horizontally. The full resolution mode should include at least four colors, all selectable from a palette of 16 colors.

The "medium-resolution" mode, call it *half-resolution*, would be nice at 350v by 360h and should offer all 16 colors. Medium resolution would be useful this way because it would be square from a coordinate point of view, even though it probably still would not have an aspect ratio of 1 on the display.

Actually, I *would* like to see more than four colors in the high-resolution mode. Three bits per pixel (giving eight colors) is a little hard to conceive, but four (16 colors) is not. Such a scheme would require 128K of memory, not a lot by today's standards and costs.

By the way, 350 by 720 is the same resolution currently generated on the IBM monochrome display, so good quality text should be possible to achieve on such a system, eliminating the need for two adapters and two displays and thus providing a more cost-effective solution than does the current PC.

PCTJ: IBM doesn't offer many graphics programs. Can it really be competitive in this market?

Fastie: If IBM builds the right hardware for the PC, the aftermarket will provide all the software, peripherals, and support that the end-user will need. A lot of that is now in place for the PC, even given its current limitations.

PCTJ: So you're still positive about the PC in general and about its utility in the graphics market?

Fastie: Yup. Although the machine is viewed by many as a conservative purchasing decision and not the best technology, it is more evident than ever that the IBM PC is the desktop computer of choice.

What do you get when you cross 1200 baud, free on-line time, and extra features at a price Hayes can't match?

Shopping for a modem doesn't have to be a riddle. A tedious sifting through claims and counterclaims. A quest for the best that raises more questions than it answers.

The new MultiModem, from Multi-Tech Systems, gives you the right answers from the start. The answers to all your questions about what a first-rate intelligent modem should do for you.

MultiModem.

Data Transmission Speed?

The MultiModem gives you a choice—either 1200 or 300 bits per second. 1200 for fast, efficient communications and lower long-distance charges. Like when you're downloading data from the corporate computer, or swapping files with a friend across the country. 300 bps for your less demanding applications, like checking out bulletin boards, playing games, or having on-line keyboard conversations.

Free On-Line Time?

With the MultiModem you get CompuServe's DemoPak—a free two-hour demonstration of their service, and up to seven more free hours of on-line time if you subscribe. You also get a $50 usage credit from NewsNet, a service which lets you tap into 150 different specialized business newsletters.

Intelligence? Extra Features?

Of course, the MultiModem automatically dials, answers, and disconnects. But it does a few extra things too, things the Hayes Smartmodem 1200™ doesn't. Like recognize dial tones and busy signals, so the MultiModem will automatically redial, or automatically try a different number. And it remembers phone numbers too, up to six of them, in its battery-backed memory.

Trademarks*—MultiModem, MultiCom PC: Multi-Tech Systems, Inc.—CompuServe: CompuServe Information Services, an H & R Block company—NewsNet: NewsNet, Inc.—Smartmodem: Hayes Microcomputer Products, Inc.—Crosstalk: Microstuf, Inc.—Data Capture: Southeastern Software—PC-Talk III: Freeware/The Headlands Press.*

Hayes-compatibility?

Yes, the MultiModem is Hayes-compatible. (Most software requires modems that are.) That means the MultiModem will run with popular communications programs like Crosstalk XVI, Data Capture, PC-Talk III, our own MultiCom PC, and dozens of others.

What About Price?

At $549, the MultiModem retails for $150 less than the Smartmodem™. You get Hayes-compatibility and so much more—for so much less.

What do you get when you cross all these things? The MultiModem. From Multi-Tech Systems—a company that's been making top-quality modems for over thirteen years.

Isn't this the answer you've been looking for?

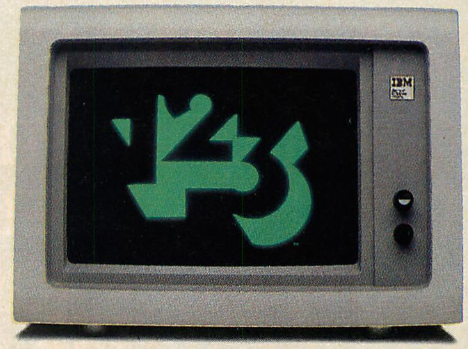
CIRCLE NO. 204 ON READER SERVICE CARD

For the name of your local distributor, write to **Multi-Tech Systems, Inc.** 82 Second Avenue S.E. New Brighton, Minnesota 55112. Or call us at (612) 631-3550.

MultiTech Systems

The right answer every time.

HERCULES™ is the only graphics card that can run 1-2-3™ on IBM's® monochrome display.



And that's just for starters.

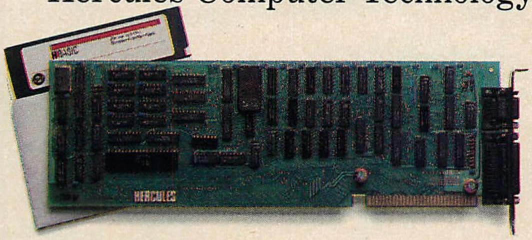
You don't have a Hercules Graphics Card? Then unfortunately you won't be able to run 1-2-3, including all its graphics, on IBM's monochrome display. And you'll miss out on all the other reasons why there are more Hercules Graphics Cards producing more high resolution graphics than any other add-on card for the IBM PC.

But don't take just our word for it. If you need convincing, remember that most of the IBM PCs at Lotus™ are running Hercules Graphics Cards. And the authors of 1-2-3 know a good card when they see one. Or consider that the Hercules Graphics Card is widely used at Rockwell, Mass Mutual, and Carnegie Mellon. They couldn't all be wrong, could they?

At **$499**, we think the Hercules Graphics Card offers the best price/performance ratio of any graphics card available today. As you can tell, plenty of users agree with us.

Call or write for our free information kit. You'll see why the first graphics card for the IBM PC is still the best.

Hercules Computer Technology, 2550 Ninth St., Suite 210, Berkeley, California 94710. Telephone: (415) 540-6000.

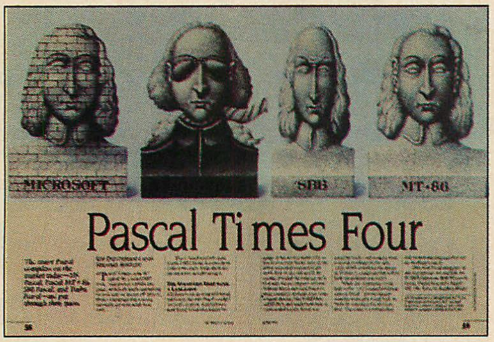


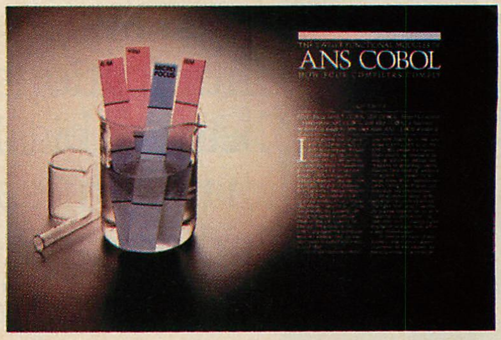
Hercules. We're strong on graphics.

CIRCLE NO. 151 ON READER SERVICE CARD

The Hercules Graphics Card offers 720 × 348 graphics resolution on IBM's monochrome display, compatibility with text mode software, a parallel printer port, software to use BASIC's graphics and a two year warranty. A graphics subroutine library with screen dump is available separately for $50. Graphics software that is compatible with the IBM color graphics card but does not have a Hercules compatible version will not run on the Hercules Graphics Card. **Foreign Distributors:** Reflex/U.K.; Computer 2000/W Germany; Edisoft/France. Hercules Graphics Card is a trademark of Hercules Computer Technology. IBM is a registered trademark of International Business Machines. 1-2-3 and Lotus are trademarks of Lotus Development.


Before IRMA™
PC-to-mainframe
communications
had advanced just
about this far.
(PHOTO CREDIT) C Aurness/West Light





Pascal Opinions

After reading your recent review of Pascal systems ("Pascal Times Four," Jeff Duntemann and Michael Bentley, July 1984, page 58), I felt an urge to express an opinion regarding Borland's Turbo. I have been using it for a few weeks and am quite pleased—not only with the performance, but also with the cost.

George Grover
Electronics for Industry
Miami, FL 33186

The poor quality of documentation and support for most microcomputer software continues to amaze me. How will the industry ever reach its potential if naive users can't be brought on-stream without wasting time on wild goose chases? Is it any wonder that [Lotus] 1-2-3 succeeds among the professional who can least afford to waste time?

I'm a new owner of Microsoft Pascal 3.2. It was claimed to be fully compatible with both the R:Base Program Interface and the 8087 on my XT. Microsoft's documentation makes compatibility with the 8087 on the XT seem less than perfect, though readily achievable with a minor patch (page 10 and Appendix C of the User's Guide). The manual plainly states that a patch is required if "your hardware configuration meets any of the following criteria: (1) It uses an 8087 interrupt vector number other than 2. (2) It uses an 8259 interrupt controller. (3) The 8087 shares interrupts with another device on the same vector." (Microsoft's FORTRAN compiler documentation contains the same wording.)

On all IBM PCs and XTs an 8087 would use interrupt vector number 0, the nonmaskable interrupt, not 2. Similarly, they all have an 8259. On mine, if it used 2, it would conflict with my Microsoft Mouse, which in turn cannot use 5 where it would conflict with the hard disk.

Since until yesterday I hadn't had to know much about interrupts, I wanted to confirm my understanding with someone. The Programmer's Shop was helpful in attitude, but not in detail. They wished me luck in dealing with Microsoft. Three calls to Microsoft and 75 minutes on hold brought me the claim that I didn't have a problem because the documentation was in error.

Can you clarify whether Microsoft's documentation or its support staff is right about the need to patch its Pascal 3.2 to use the 8087 properly on an IBM PC or XT? *PC Tech Journal* and your article on Pascal compilers have been sources of good information for me in the past. Thanks for your help.

Dennis C. During
Brooklyn, NY

Microsoft's documentation is in error on this point. MS Pascal 3.2 needs no patching to operate correctly with the 8087 coprocessor. Benchmarks were compiled with MS Pascal exactly as it was received from Microsoft without patches.
—Jeff Duntemann

Thank you for your insights into the various Pascals, even if they did nothing more than confirm my choice of MT+ for large overlayed programs.

I question your result of "failed" for the pentathlon floating-point test with the 8087. I tried on my machine (a Compaq Plus with 8087), and it seemed to work. I say seemed because the only criterion used in your benchmark is that the procedure bench1 ran to completion, not that it produced a certain numerical result. Is this a mistake?

It's a good thing that you did not review version 3.1 of MT+ or you would have had the result "failed" for all of the 8087 test; the 8087 library just flat out did not work.

Also, you did not mention the compatibility of MT+ with the other DRI compilers both for the 8086 under PC-DOS and CP/M 86 and CP/M 80. Although this might not seem important for readers of an exclusively PC magazine, I think it is worth noting for those who want to write software that runs on a variety of operating systems.

Hardy Tichenor
San Rafael, CA

That particular benchmark (floating-point test) failed in that it locked up the machine and forced us to reboot. We ran the code again, and it failed in the same fashion.

Why your compilation worked while ours did not could be due to a number of different reasons, but I suspect the version of MT+86 that we tested was not identical to the version 3.2 release that was put into general distribution. Our copy came directly from DRI a little ahead of the official V 3.2 release date. Perhaps a leftover glitch from V 3.1 remained in that version.

On your point of compatibility, as we have little or no experience with other DRI compilers, such as CB 80, CB 86 and DR C, we did not feel we should comment. We do know, however, that MT+86 is extremely compatible with MT+80 and forms a good downward path from the IBM PC world to the eight-bit world of CP/M 80.
—Jeff Duntemann

The Rest of the Story?

The report on the four COBOL compilers ("The Twelve Functional Modules of ANS COBOL: How Four Compilers Comply," Casey Pontius, April 1984, page 76) was an excellent article. However, following page 82 I could not find any closing comments or summary ended with the little PC system unit symbol anywhere in that issue. I infer thereby, that the article as printed was incomplete, and I have been looking for a follow-up in subsequent issues, with

The truth about information management:

KnowledgeMan™ beats dBASE® over 250 ways.

"...the leading seller is not the best."

You get more from KnowledgeMan. Relational data management, ad hoc queries (like IBM's SQL/DS), spreadsheet, math and statistics, printed forms management, screen I/O management, a full-scale programming language. **Plus** you can now get a text processor, a forms painter, color graphics and a run-time version.

All completely integrated. So you can do just about anything you wish. Like financial modeling, order entry, job costing or decision support.

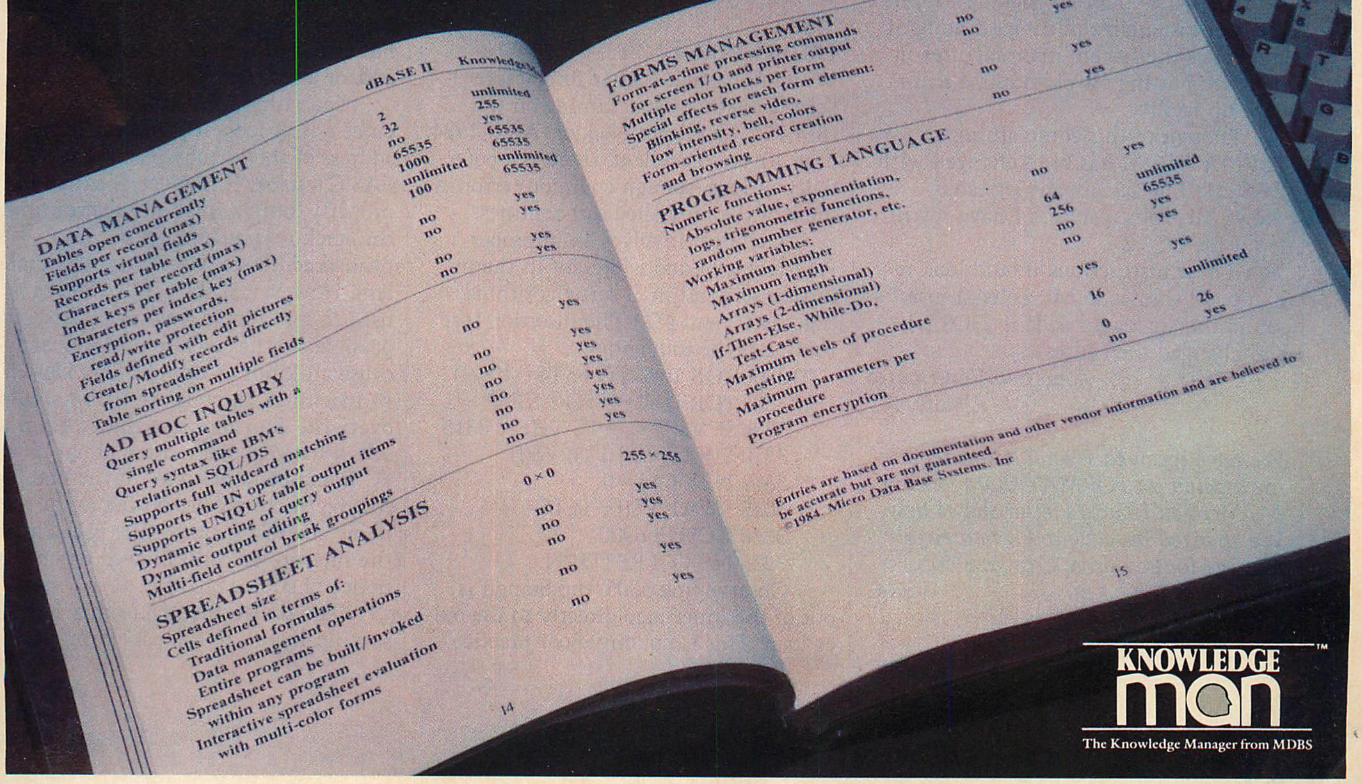
If you'd like to see more about how KnowledgeMan beats dBASE, see your dealer today and ask for a free feature-by-feature comparison. Or call or write: Micro Data Base Systems, Inc., P.O. Box 248, Lafayette, IN 47902, (317) 463-2581.

Operating Systems: PCDOS, MSDOS, CP/M-86. Requirements: 192K RAM
Text processing, forms painting and color graphics are optional integrated KnowledgeMan components.

KnowledgeMan is a trademark of Micro Data Base Systems, Inc., dBASE II of Ashton-Tate, SQL/DS of IBM. Current release is 1.06 as of 4/1/84. Color graphics requires PCDOS and IBM color graphics card or equivalent.

CIRCLE NO. 118 ON READER SERVICE CARD

John—
"K-man" not only looks superior to dBASE; I've checked it against others & it seems to be a generation or two ahead!
J.P.



KNOWLEDGE man™
The Knowledge Manager from MDBS

no success as yet. Am I looking for the proverbial "needle in a haystack," or is the summary still to come?

Jeff Davidson

Sorry for the confusion. We discovered the loss of the "end slug" only after the issue went to press. Not to bore you with publishing details, but that end slug is something that has to be added manually (that is, it cannot be set by the typesetting computer), and somebody obviously goofed.

That was the end of the article. We specifically avoided conclusions in that piece because we were trying to report on ANSI compliance, not the nature of each particular product. We are working now on reviews of the major products.

Thanks for your note.

—WF

A Patch to Match 2.1

James Folts' piece on "Disarming DOS FORMAT" (July 1984, page 32) presents an excellent way to obviate a real danger. I know of two misFORMATs that wiped out hard disks at my firm. Giving computer power to nontechnical users has created a whole industry; it is, however, unfortunate that much of the resultant system is not ready for these users. The ability to erase megabytes of information inadvertently and easily while attempting an often-used and necessary utility is one example of this; another case is the near certainty that the sad user with the reformatted disk has not backed up recently. A back-up requiring much time spent shuffling diskettes is worse than no back-up—it lets the user rationalize putting off doing it, and it lets the system manufacturer rationalize putting off providing a real back-up facility.

Enough of this complaining. FORMAT.COM has been changed with DOS 2.1. The accompanying table shows Mr. Folts' patch as I have rewritten it for DOS 2.1.

These instructions should look remarkably similar to Mr. Folts'; I just found the analogous code in DOS 2.1 and changed the addresses.

Jonathan Van Houtte
Albany, NY

We have received several letters requesting a patch for FORMAT.COM version 2.1. Mr. Van Houtte's letter shows how. We changed his code to give an error message for any drive C or greater.

—SG-C

DOS 2.1 FORMAT Patch

Type DEBUG FORMAT.COM (enter).
Check your FORMAT utility to be sure it is the correct code for this patch:

```
-U 19E
xxxx:019E  BA5C09   MOV DX,095C
xxxx:01A1  B425     MOV AH,25
xxxx:01A3  B023     MOV AL,23
xxxx:01A5  CD21     INT 21
```

Change the instruction at 19E so that we can add the patch:

```
-E 19E
xxxx:019E  BA.E8 5C.1F
xxxx:01A0  09.18
```

Dump from location 19C0 to make sure it is zeros:

```
-D 19C0
xxxx:19C0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
xxxx:19D0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
xxxx:19E0  00 00 00 00 00 00 00 00-00 00 00 00 00 00 00 00 ................
```

Now assemble the trap:

```
-A 19C0
xxxx:19C0 CMP AL,2;is fixed disk drive C?
xxxx:19C2 JNL 19C8 ;print error message
                   ;if drive >=C.
xxxx:19C4 MOV DX,95C
xxxx:19C7 RET
xxxx:19C8 MOV DX,FAF
xxxx:19CB JMP 4FB
xxxx:19CE ^C
```

Dumping from 19C0 should show:

```
-D 19C0
xxxx:19C0  3C 02 74 04 BA 5C 09 C3-BAAF 0F E9 2D EB 00 00<}.:\.C:/.i-k....
xxxx:19D0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00......
xxxx:19E0  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00......
-W
-Q
```

Improper Exit

While the article "After Bubblesort: Sorting Methods and Timing on the IBM PC" by Eliezer Naddor (April 1984, page 40) was excellent, there is a potentially serious (though common) error in some of the published sort routines.

The error involves an improper FOR loop exit and is present in the insertion sort and shell sort algorithms (figures 4C and 4D.) The insertion sort routine is the following:

```
3300 FOR I=2 TO N:T$=V$ (I)
3305 FOR J=I-1 TO 1 STEP -1
3306 IF V$(J) <=T$ THEN 3315
3310 SWAP V$(J+1), V$(J)
3312 NEXT J:J=0
3315 SWAP V$(J+1),T$
3320 GOSUB 6900
3325 NEXT I:RETURN
```

Observe line 3305: the branch is out of the inner loop directly to the outer loop. This is not only poor practice, but also it can result in damaged data and/or a program crash.

Each time Microsoft BASIC executes a FOR loop, it stores the beginning address of the loop by pushing it onto BASIC's stack. If the loop is exited improperly, only a small number of times, the stack is large enough so these pushed addresses will not cause a problem. However, if the SORT routine is used to sort a large amount of data, it is possible for the stack to overflow and cause all kinds of trouble. On older CP/M machines, the stack overflow would overwrite data and even the program itself; what gets overwritten on an IBM PC is an open question. Maybe nothing. Maybe Microsoft has bowed to the inevitable and taken care of the problem in the interpreter. Still, I haven't seen any notice of such a change.

The BASIC manual (somewhere) specifically warns against exiting FOR

loops in this way; the WHILE/WEND construction was added to the BASIC interpreter and compiler by Microsoft specifically because of this problem. Moreover, WHILE/WEND loops may be exited in the "improper" manner without any similar problems because the return address is not placed on a stack.

This particular way of coding loops is common, and your publication is neither the first nor, I fear, the last to print a program using it. Please warn the author and your readers against using such dangerous and illegal methods. If the BASIC manual specifically warns against doing something, it is usually with good reason.

Bruce W. Tonkin
Round Lake, IL

Mr. Tonkin's complaint is not supported by the BASIC manual. On the contrary, on page 4-89 the manual actually speaks of branching out of a loop.
—Eliezer Naddor

TWO B: OR NOT TWO B:

In "Nested Batch Files" (November/December 1983, page 223) Jeff Garbers described how a secondary command processor can be used to run nested batch files. I have just been trying unsuccessfully to apply this to a pair of batch files that are on different drives.

From Garber's article and, more particularly, from page 10-9 of the PC-DOS 2.0 manual, I am led to expect that the line

COMMAND B:/C B:FILE2

appearing in A:FILE1.BAT should hand over control to FILE2.BAT on DRIVE B. Yet with this and with every plausible variation I have tried, I get the message, "Specified COMMAND search directory bad," before being rather abruptly ushered out to the DOS prompt.

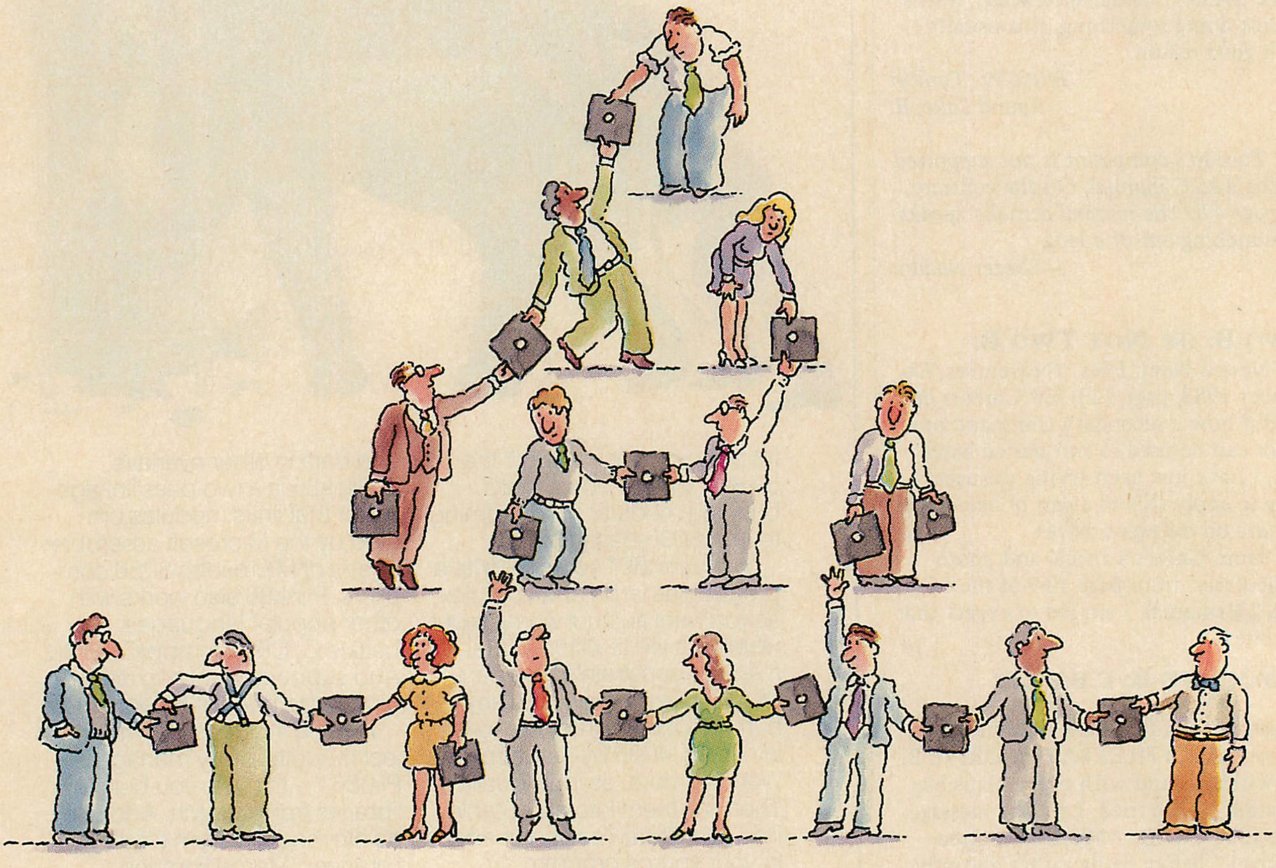
What's wrong here?

Sid Socolar
New York, NY

Mr. Socolar's problem is my favorite kind: it's both interesting and easy to solve. All he needs to do is leave out the first "B:", leaving his command line as "command /C B:FILE2". This seems to work quite well.

His approach fails for two reasons. According to the page he cites in the DOS manual, the parameter appearing before the /C in the COMMAND line specifies "the directory searched for the command processor to be loaded". So Mr. Socolar's command line would tell

THE END OF SOFTWARE FREEBIES.



Finally there's a foolproof way to protect software against unauthorized duplication. And the technology is all on the disk itself.

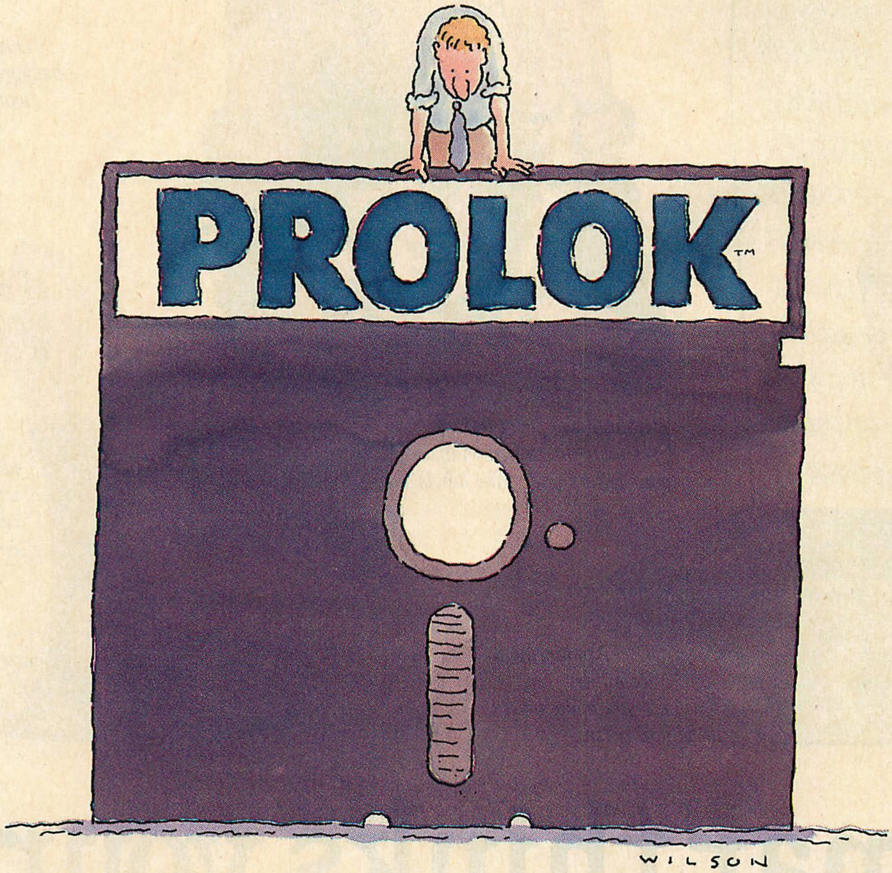
The new Prolok™ disk doesn't need add-on hardware. Instead each diskette is marked with a unique, physical "fingerprint." No two are alike. A precise description of the individual print is encoded magnetically. The fingerprint AND the description must match exactly before the software is decrypted and released to the system. No match, no access.

Its genius is its simplicity and familiarity. Prolok looks like an unprotected disk, loads like an unprotected disk, works like an unprotected disk. The user feels immediately at home and in command.

Backups are easily made via normal system utilities. However, to be read they must be accompanied in the system by the original Prolok disk.

Prolok puts the casual copier—and even the deliberate pirate—out of business. It barely

PROLOK. SOFTWARE



increases the price of your product, yet it makes sure your customers don't buy one program and copy ten.

Several command line slash (/) options are built into Prolok diskettes for customized security, depending on your needs.

Software can be loaded easily onto Prolok diskettes using any system from a PC to commercial mass duplication equipment.

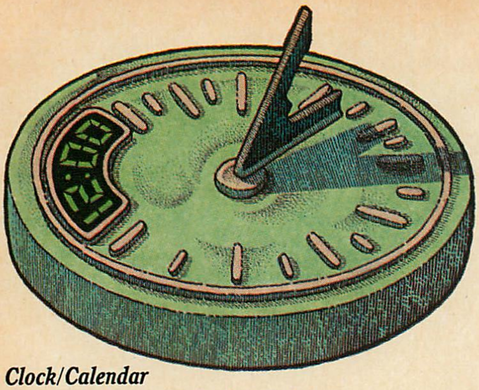
Prolok is an engineering breakthrough of Vault Corporation, which has been successfully safeguarding software since the inception of security disk technology. Over 2000 businesses and organizations protect their valuable programs with Prolok.

Simply contact Vault Corporation at 2649 Townsgate Road, Suite 500, Westlake Village, CA 91361. Or phone us at 800-445-0193 (U.S.) or 800-821-8638 (California). And find out why software freebies are becoming a thing of the past.

PROLOK™

PROTECTION, RIGHT ON THE DISK.

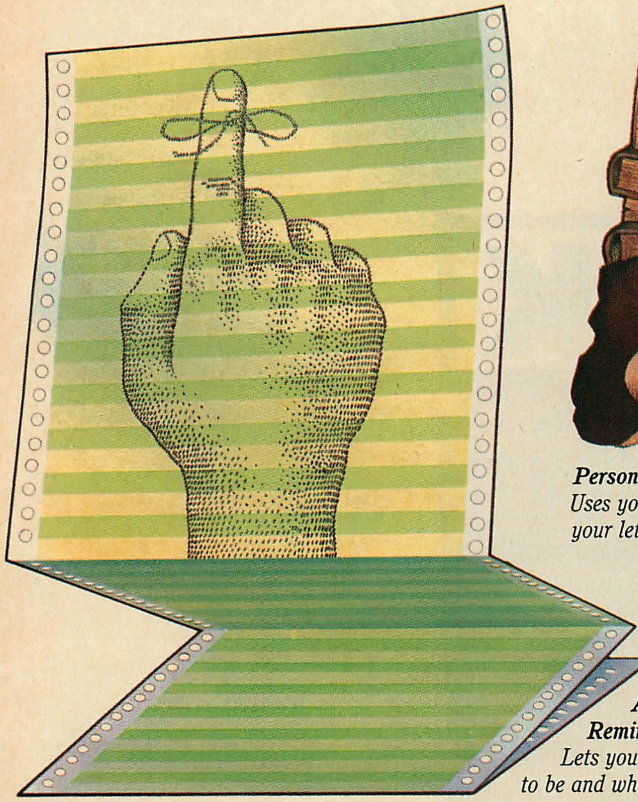
CIRCLE NO. 180 ON READER SERVICE CARD



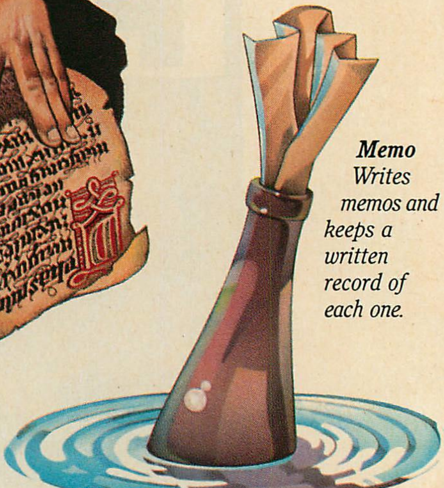
Clock/Calendar
Automatically inserts the date and time when you turn on your computer.

Encode/Decode
Ensures the security of your files.

Calculator
Does your figuring in background mode so you won't have to change programs.



Personalized form letters
Uses your file to personalize your letters and make labels.



Memo
Writes memos and keeps a written record of each one.

Appointment Reminder
Lets you know where to be and when.

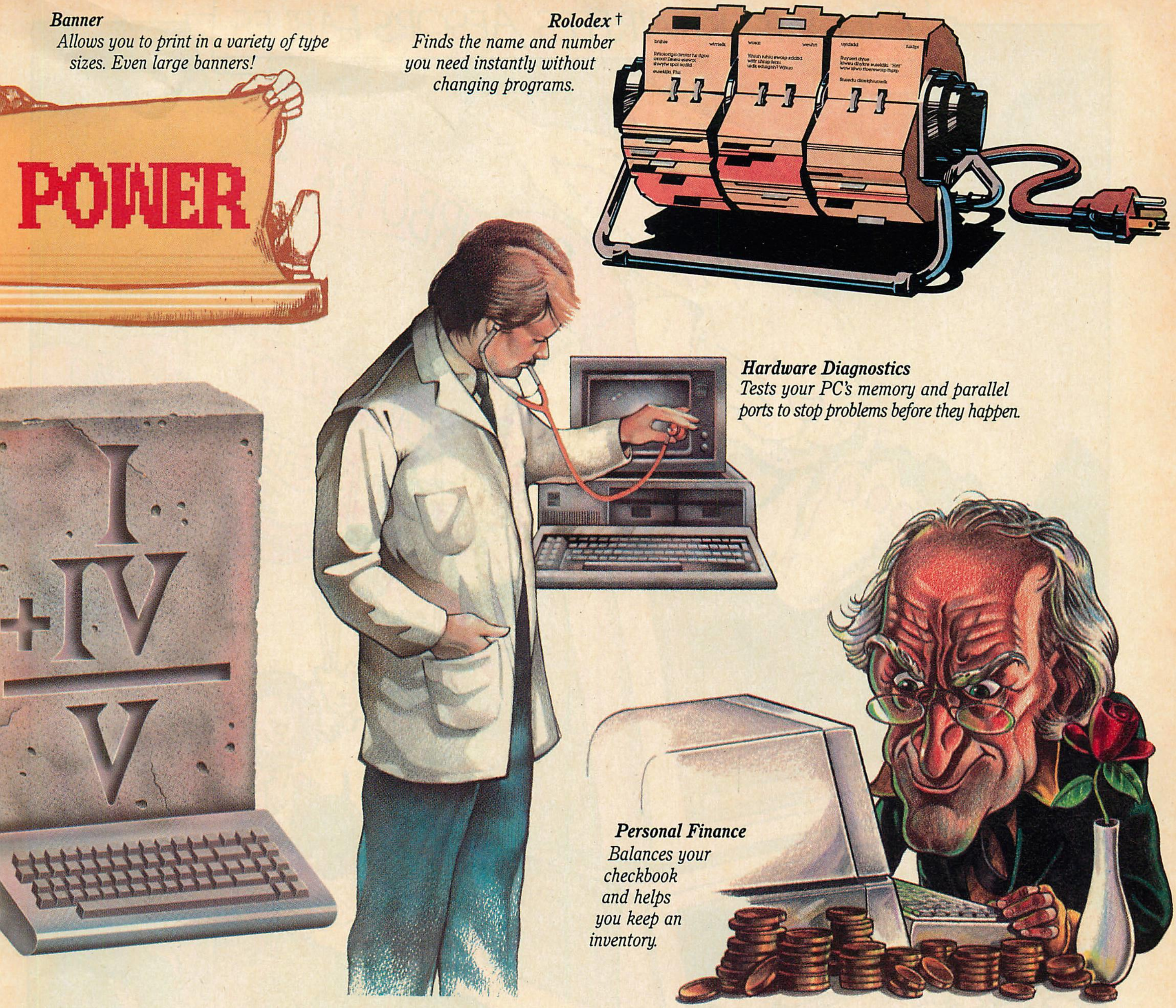
Tecmar thinks your PC

As your business grows, so should your computer. Only Tecmar offers you this unique package of hardware and free software. With it your PC can grow to its full potential.

Tecmar Multifunction Boards...
"extremely powerful."—PC World
A Tecmar board will expand your computer's memory and increase its speed and efficiency. It will give you additional ports to accommodate a wide range of peripherals from modems and plotters to dot matrix and letter quality printers.

With a Tecmar board you can run powerful programs like Lotus 1-2-3™* and WordStar®†† on a PCjr. The RamSpooler makes printing a background task and frees your PC for other jobs. A built-in clock/calendar automatically inserts the date and time at power on and is independently powered by an easily replaced battery.

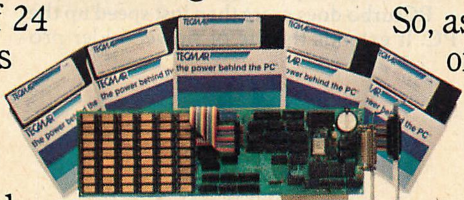
Every Tecmar multifunction board is run through a series of rigorous tests to ensure quality. Our incredibly low failure rate (0.4%) is unparalleled. All boards are additionally backed by a full one-year warranty.



should do all of the above.

Free Software *"...a chest of Jewels."—PC Week*
Great hardware deserves great software. So, if you buy a Tecmar board we'll give you our Treasure Chest™ of Software at no extra charge. The Treasure Chest consists of 24 programs that include business applications, a calculator, a security system, hardware diagnostics, even games! Most of these can be run in background mode with programs like Lotus 1-2-3 and WordStar. Using these features is as easy as a couple of keystrokes, and without changing disks. No other company offers you such an extensive array of software with their multifunction boards.

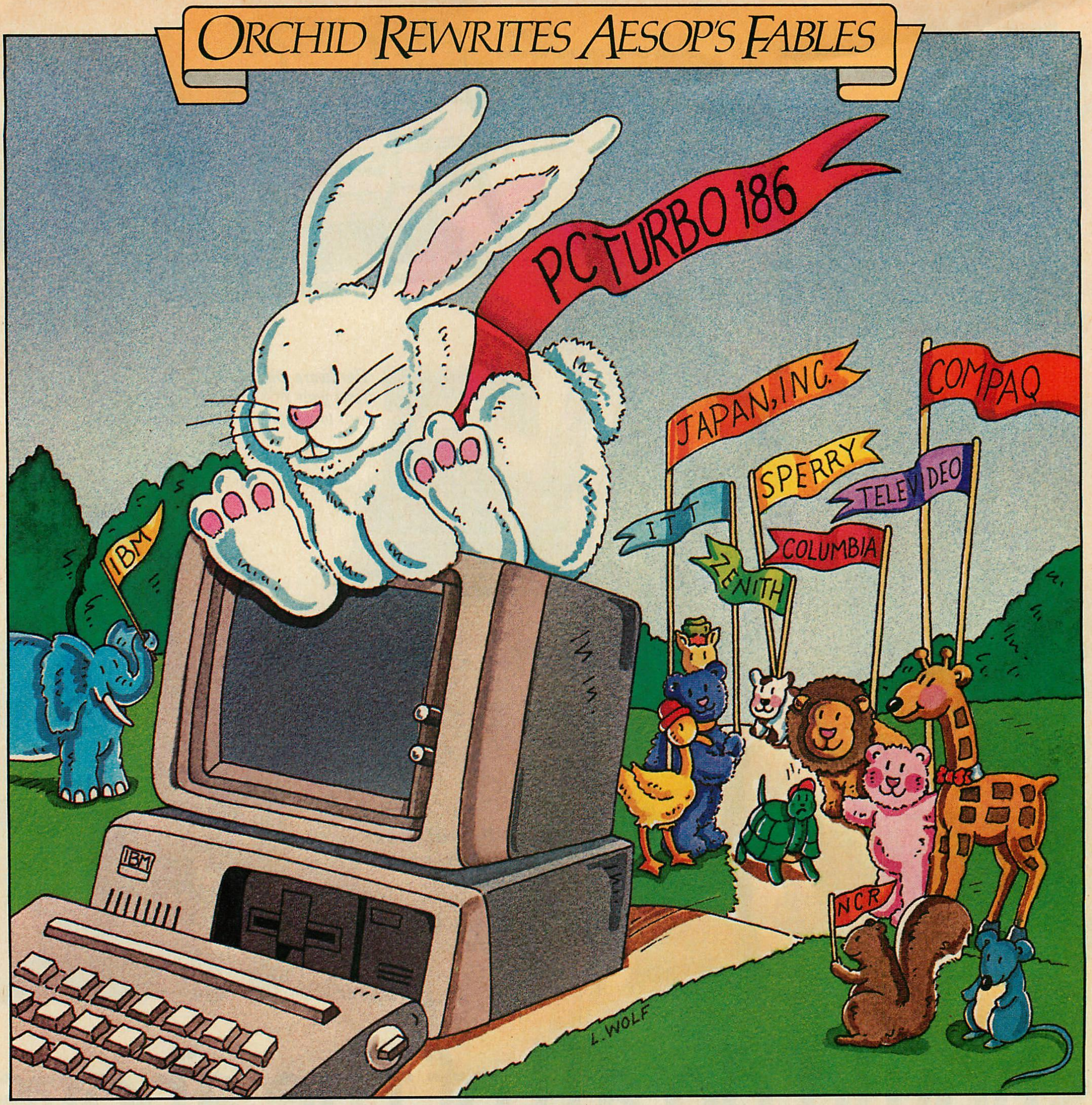
So, ask your dealer for a demonstration of any of Tecmar's multifunction boards. And check out the free software while you're there. Or call 216-349-0600 for the dealer nearest you.



Tecmar
THE POWER BEHIND THE PC

°Lotus 1-2-3 is a registered trademark of the Lotus Development Corporation. †Rolodex is a registered trademark of the Rolodex Corporation. ††WordStar is a registered trademark of the MicroPro International Corporation.

CIRCLE NO. 195 ON READER SERVICE CARD



PC TURBO 186...Speed Wins the Race.

In business, success comes to those who win the race against deadlines and competition. The IBM™ PC is a great tool, but valuable time can be wasted waiting for it to recalculate spreadsheets, retrieve data, or execute the newer and more complex software packages. PCturbo,™ new from Orchid, can help you win and enjoy the fruits of success.

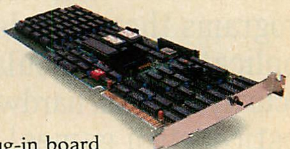
PCturbo is the ultimate IBM PC productivity enhancement. It boosts the execution speed of your PC so you can get more done in less time. And it's transparent to existing programs such as WordStar,™ Lotus 1-2-3,™ and dBase II.™ So, with PCturbo, your PC looks and acts the same as before; it just runs faster.

Who needs PCturbo? Anyone who ever waits for their PC to finish executing so they can run something else. Programmers waiting for compilers to finish before testing a new feature. Businessmen waiting for a data base to retrieve account information. Word processors waiting for the spelling checker to finish before printing.

PCturbo does more than just speed up the PC. It extends the life of slower, older programs. It provides a base for new sophisticated software. And it protects your PC investment. You get your work done quicker and decisions made sooner. The very reasons you got a computer in the first place. Write or call for more information.

ORCHID

ORCHID TECHNOLOGY
47790 Westinghouse Drive
Fremont, CA 94539
(415) 490-8586 Telex: 709289



TECHNICAL DETAILS:
Hardware
- One slot plug-in board with high-speed processor (80186)
- Up to 640K on-board memory for a maximum of 1.28 Mbytes total PC memory.

Software
- Runs IBM PC-DOS.
- Provides disk caching, RAM disk using available PC memory up to 640 Kbytes.

CIRCLE NO 227 ON READER SERVICE CARD

PCturbo is a trademark of Orchid Technology. IBM PC and XT are trademarks of International Business Machines Corporation. WordStar is a trademark of MicroPro. Lotus 1-2-3 is a trademark of Lotus Development Corporation. dBase II is a trademark of Ashton-Tate.

DOS to look on Drive B: for the command PROCESSOR, not the command to be processed. If he leaves that optional argument blank as I suggest, DOS will look for the command processor in the path specified by the environment's COMSPEC variable.

The second problem is more subtle. Without a space between the first B: and the /C, COMMAND's parser doesn't interpret the /C as a switch. Instead, it apparently takes "B:/C" as an indivisible string, and tries to interpret it as the directory to be searched for the command processor. This could have something to do with the fact that under certain undocumented conditions, DOS will accept the forward slash as well as the backward slash as a path name delimiter.

—Jeff Garbers

PNP UPDATE

There is a conflict between Peeks 'n Pokes (PnP) and the Hercules Graphics Card. The Hercules card uses the same control port (952 or 3B8H) as the IBM monochrome card, but it uses a previously unused bit (bit 1) to indicate graphics mode. CRT.BAS and CRTDEMO.BAS in PnP set this bit when disabling the screen and blink. This can damage the monochrome monitor if the Hercules card is used.

To eliminate the conflict, we have a new version of PnP, numbered 2.1. To update your disk, replace the lines in the two programs as shown below. Note that this fix does not work for the Hercules graphics mode.

CRT.BAS (change three lines)

```
260 OUT 952,41 ' Turn monochrome
    display back on
300 OUT 952.9 ' disable monochrome
    blink
320 OUT 952,41 ' enable monochrome
    blink
```

CRTDEMO.BAS (change four lines)

```
50 IF (PEEK (16) and 48)=48 THEN
    DEF SEG=45056!:
OUTBUFF=952:GOTO 70 ' mono-
    chrome
60 DEF SEG=47104!:OUTBUFF=
    984:SCREEN 0 ' color
180 OUT OUTBUFF,9
210 OUT OUTBUFF,41
```

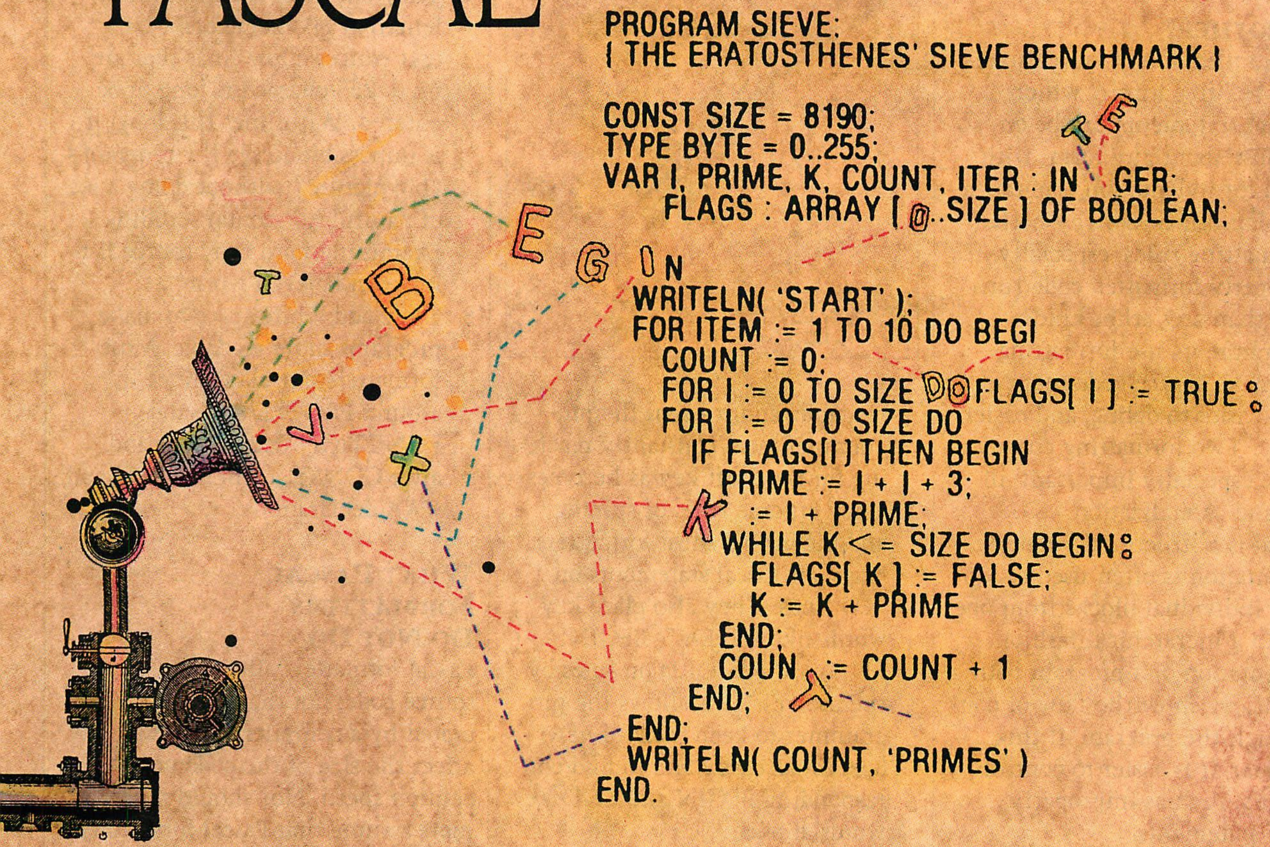
After making the above changes, change line 40 in the program MENU.BAS to display version number 2.1

Brett Salter
Data Base Decisions
Atlanta, GA

ILLUSTRATION • DAVID POVILAITIS

Steven Armbrust and Ted Forgeron

THE WELL-FORMATTED PASCAL



Two Pascal formatting utilities: TIDY is tidier than PASCAL UTILITIES, but either of these packages is worth having.

Everybody loves reading well-formatted Pascal code, but after five or six revisions of a program, most people don't really feel like fixing all the BEGIN . . . END blocks so they are indented properly. Also, because the Shift key on the IBM PC is hard to reach, it is annoying to have to type all the reserved words in upper-case letters. Most programmers really intend to go back later and make their programs prettier, but sooner or later their good intentions dwindle away.

Steven Armbrust has been a technical writer in the computer industry for more than a decade. Currently he is working as a senior documentation engineer at Multisoft Corporation in Beaverton, Oregon. Ted Forgeron is vice president of systems software engineering, also at Multisoft Corporation.

Luckily, there is a way to have well-formatted code without the frustrations of having to type it in that way. Now there are utilities, called in generic terms *Pascal formatters,* that can turn even the murkiest Pascal source code into easy-to-read, professional-looking code. This article discusses Pascal formatters—what they are and how they are useful. It also examines two products available for the IBM PC: TIDY from Major Software and PASCAL UTILITIES from Seven Valleys Software.

WHY BOTHER?

Pascal compilers allow source statements to be "free form," which means that statements can be entered in just about any form, with no regard for indentation, capitalization, or blank lines. So why bother with formatting rules, especially since properly formatted code runs no faster than any other kind?

Programmers use formatting rules to make source code more readable, easier to debug, and easier to modify. Two formatting considerations are especially important: indenting and capitalization.

Pascal is a structured language, and indentation is often used to separate the individual blocks from one another in a hierarchical fashion. The indentation of a program listing should clearly show the flow of control through the program; if this flow is clear, it is easier to control logic errors, such as improperly nested IF statements.

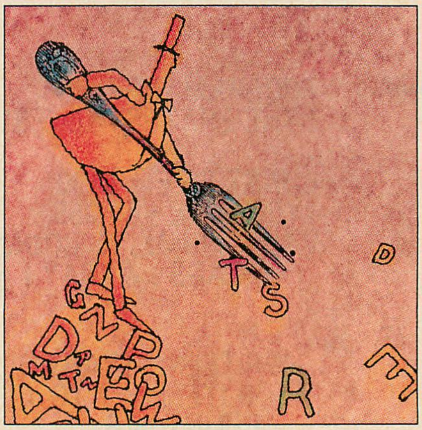
Indentation also helps to clear up complex data descriptions, such as a RECORD with many fields. Pascal records can be hierarchical, and using indentation can help keep the hierarchy straight.

Capitalization helps make the essence of the program easily visible. By capitalizing the Pascal reserved words (such as BEGIN, END, IF, THEN, ELSE, etc.), programmers can make the action words stand out against the rest of the program.

WHAT PASCAL FORMATTERS DO

Pascal formatters take source code—regardless of the formatting rules used in creating it—and turn it into consistently formatted source code. The formatting rules used by these utilities vary from product to product, but most Pascal formatters perform the following operations:

1. They transform Pascal reserved words into upper-case letters. This makes the program more readable, and it keeps the programmer from constantly reaching for the Shift key as he types.



2. They enforce rules of spacing and indenting. Most formatters begin all Pascal statements on new lines. They also place key words like REPEAT, BEGIN, END, and RECORD on lines by themselves. This allows the programmer to keep track of new operations easily and to find corresponding BEGIN and END key words at once.

   Formatters indent the bodies of BEGIN . . . END, FOR, REPEAT, WHILE, WITH, and CASE statements from the key words. They also indent the bodies of LABEL, CONST, TYPE, and VAR declarations from the declaration key words. Structures and IF . . . THEN . . . ELSE clauses are treated similarly.

   Formatters vary in the number of spaces they indent, but the best ones indent two spaces for each level of indentation. Indenting more than two spaces causes "rightward creep" when complex programs are involved (most of the code ends up on the extreme right side of the page). Indenting less than two spaces per level makes it hard to distinguish the individual levels.
3. They format comments in special ways. Most formatters adjust comments so that they begin in the same column or use a consistent amount of spacing before and after the comment.
4. They enforce a maximum line length of 80 characters. Many text editors allow lines that are up to 255 characters long to be entered. With lines this long, it is impossible to see the program's "big picture" without doing a lot of painful horizontal scrolling. Examining hard-copy source listings can be even worse.
5. They add white space to make programs more readable. Most formatters put blank lines around PROCEDURE and FUNCTION declarations and after CONST, LABEL, TYPE, and VAR declarations.

WHAT PASCAL FORMATTERS DO NOT DO

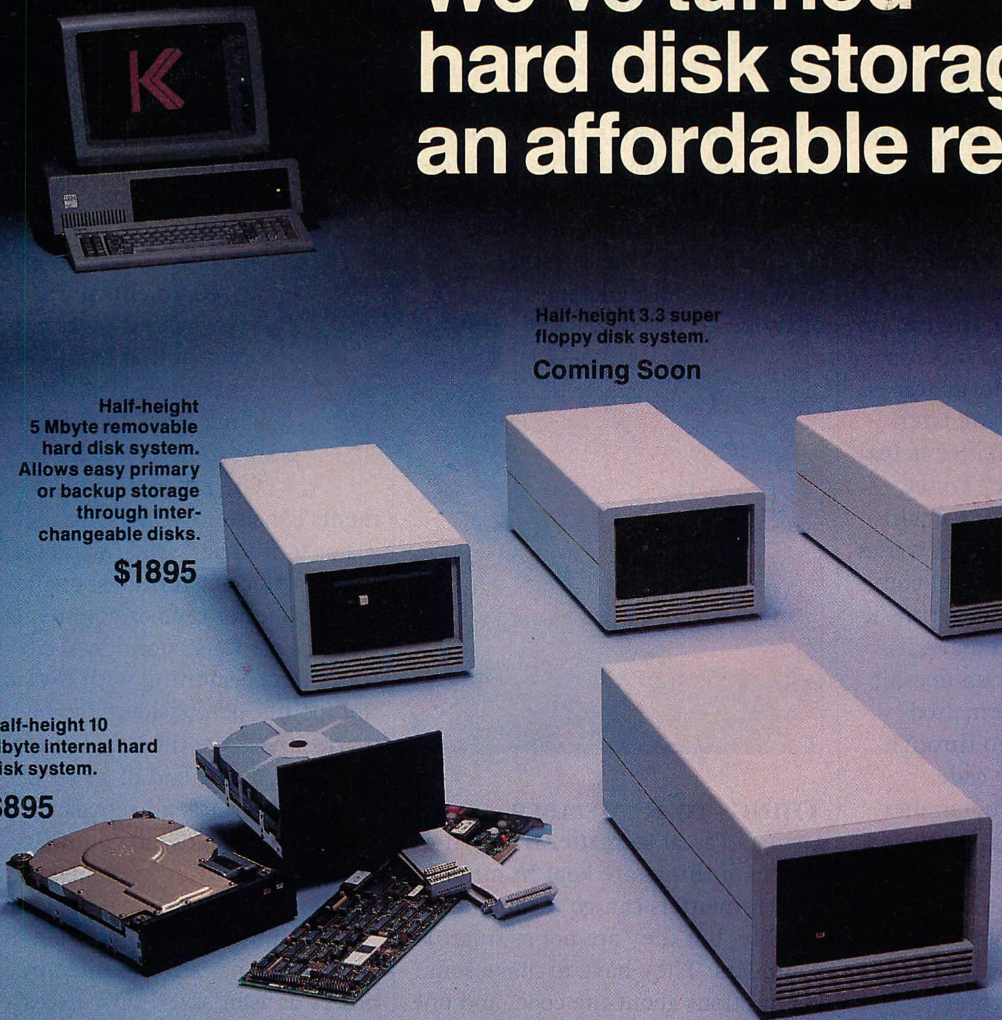
Pascal formatters affect only the source code, not the object code generated by a compiler. They will not affect the size of the code or its execution speed. Nor should programmers assume that Pascal formatters are syntax checkers—they are not. Code that will not compile before it is run through a formatter certainly will not compile afterward. In fact, using a formatter on code that contains syntax errors often increases the number of problems.

WHO SHOULD USE A PASCAL FORMATTER?

Pascal formatters are useful for programmers working on individual projects and for programmers work-

FOR THE IBM® PC, IBM® XT, COMPAQ,™
CORONA,™ CORONA™ PORTABLE, LEADING EDGE,® EAGLE PC,® TAVA.™

We've turned hard disk storage into an affordable reality.



Half-height 3.3 super floppy disk system.
Coming Soon

Half-height 5 Mbyte removable hard disk system. Allows easy primary or backup storage through interchangeable disks.
$1895

Half-height streamer tape system. Ideal data backup system for large-scale storage.
$1095

Half-height 10 Mbyte internal hard disk system.
$895

Half-height external hard disk system (10, 20, or 33 Mbyte). Also accommodates half-height backup system.
$1295-$2295

Kamerman Labs combines the speed and storage capacity of hard disk technology with prices you can't afford to pass up.

A dramatic leap in performance for your personal computer.
Even the smallest Kamerman Labs hard disk unit gives you 30 times the storage capacity of a typical floppy disk. Sophisticated business and professional data storage requirements can now be handled with ease.

Better yet, you can access data up to 10 times faster than with floppies. Spend more time working and less time waiting.

And it's easy to use. Kamerman Labs lets you boot directly off the hard disk, just like the IBM XT.

A complete backup solution.
Backup of hard disk data is a critical requirement in many microcomputer applications, so Kamerman Labs offers several reliable and cost-effective backup devices that are fully compatible with our hard disk units.

Lower prices than ever.
Because you deal directly with Kamerman Labs, we offer you remarkably low prices. Finally, you can realize the full benefits of hard disk technology without the excessive costs.

Tested and formatted.
All Kamerman Labs systems have been thoroughly tested and formatted before delivery.

Full warranty and technical support.
Kamerman Labs backs all its products with a full one-year warranty on both parts and labor. Plus you get prompt technical support through a toll-free hotline number.

To order, call toll-free 800-522-2237.
In Oregon, Hawaii and Alaska call **503-626-6877.**

All Kamerman Labs units come in either internal or external configurations.

IBM is a registered trademark of International Business Machines Corp.
Compaq is a trademark of Compaq Computer Corp.
Corona is a trademark of Corona Data Systems.

Leading Edge is a registered trademark of Leading Edge Products, Inc.
Eagle is a registered trademark of Eagle Computers.
TAVA is a trademark of TAVA Corporation.

Kamerman Labs

8054 S.W. Nimbus, Bldg. 6, Beaverton, Or 97005
Phone 503-626-6877

DEALER INQUIRIES CIRCLE 200 ON READER SERVICE CARD

CIRCLE NO. 225 ON READER SERVICE CARD

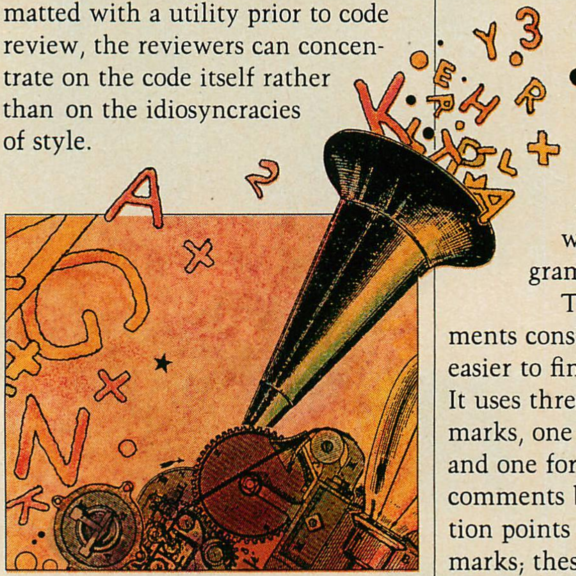
ing as part of a team. Individual programmers benefit in several ways: first, using a Pascal formatter allows them to key in their code much more quickly than they would be able to if they were formatting it themselves. After all, the Shift key is hard to hit when blazing out code at high speeds, and complex programs require too much indenting to enter quickly (not to mention having to think about how much to indent). Programmers can type their code in lower-case letters and can be lazy or inconsistent about indentation if they desire. The formatter will take care of everything later.

Second, using a formatter helps programmers debug code. To see how, consider the following chain of events. First, a programmer writes a Pascal program and gets it to compile without errors. Then he runs it through the formatter to make it easy to read. If the program doesn't execute correctly, the formatted source code is easier to go through than the original, sloppy code.

Now suppose the programmer wants to change the program to add new features or fix old bugs. He inserts his code changes into the formatted source file, but types them at the left margin, in lower-case characters. It's easier to enter changes that way, and the changes stand out against the perfectly formatted remaining code.

Next he compiles his code, but he doesn't run it through the formatter yet. Instead, he tests the code first, to see if he has introduced any new bugs. If so, he can flip through a listing and easily spot the changes made since the last time the program worked. Once all the bugs are fixed, he can run the program through the formatter again.

A programming team gets the same benefits from using a formatter that an individual programmer does, plus one more—consistent coding style. Most programmers agree that proper formatting is important, but few agree on which format is best. If everyone's code is run through a formatter, individual programmers can maintain their own unique coding habits, yet the final product is consistent. This is especially important for a programming team that uses code reviews as a means of checking the accuracy of code. If all of the code has been formatted with a utility prior to code review, the reviewers can concentrate on the code itself rather than on the idiosyncracies of style.



THINGS TO REMEMBER

Using a Pascal formatter is easy if a couple of things are kept in mind: first, formatters can go through code quickly, but they are not as smart as compilers. They have to make some assumptions about the code, and one of the assumptions is that the code is free of compilation errors. If the program contains syntax errors, a formatter either will produce strange-looking output or (if the program is bad enough) will return an error message without producing any output at all.

Second, programmers should always save their original source code until they are sure the output produced by the formatter is correct. An inadvertent syntax error might cause the formatter to abort without producing an output file.

TIDY

TIDY is a lightning-fast Pascal formatter from Major Software of Los Altos, California. It's inexpensive ($39.00), works with any version of DOS, and can format any programs written in IBM Pascal, Microsoft Pascal, or Turbo Pascal. It will run on any IBM PC (or compatible) that has at least 128K bytes of RAM.

TIDY has all the formatting features listed earlier: it capitalizes key words, enforces rules of spacing and indenting (indenting two spaces for each level), formats comments in special ways, enforces a maximum line length of 80 characters, and adds white space to make programs more readable.

The product formats comments consistently, making them easier to find and read in the listing. It uses three conventions: one for remarks, one for regular comments, and one for long comments. All comments beginning with exclamation points are considered to be remarks; these are the comments that can occur at the end of a line of code. TIDY starts all remarks at column 45 (or as close to column 45 as possible), regardless of where they are found in the original program.

Short comments are single-line comments that are delimited by braces or asterisks. TIDY places short comments on separate lines, no matter where they are placed in the original program. If the comment is too long, TIDY breaks it up into multiple lines.

Long comments are comments delimited by braces or asterisks that extend beyond a single line. TIDY starts a long comment on a new line, but it does not rearrange the text of the comment. This feature allows programmers to put comment delimiters around code that they don't want to compile but still want to save for reference.

Formatting can be turned off and on by including comments of the form {*off*} and {*on*}.

The documentation for TIDY consists only of a six-page listing

COHERENT™ IS SUPERIOR TO UNIX* AND IT'S AVAILABLE TODAY ON THE IBM PC.

Mark Williams Company hasn't just taken a mini-computer operating system, like UNIX, and ported it to the PC. We wrote COHERENT ourselves. We were able to bring UNIX capability to the PC with the PC in mind, making it the most efficient personal computer work station available at an unbelievable price.

For the first time you get a multi-user, multitasking operating system on your IBM PC. Because COHERENT is UNIX-compatible, UNIX software will run on the PC under COHERENT.

The software system includes a C-compiler and over 100 utilities, all for $500. Similar environments cost thousands more.

COHERENT on the IBM PC requires a hard disk and 256K memory. It's available on the IBM XT, and Tecmar, Davong and Corvus hard disks.

Available now. For additional information, call or write,

Mark Williams Company
1430 West Wrightwood, Chicago, Illinois 60614
312/472-6659

CIRCLE NO. 164 ON READER SERVICE CARD

Mark
Williams
Company

COHERENT is a trademark of Mark Williams Company.
*UNIX is as trademark of Bell Laboratories.

CIRCLE NO. 186 ON READER SERVICE CARD

IBM PC USERS:

TeamMate is breaking installation and performance records.

Let the TeamMate 1000 Series of Winchester and Kodak 3.3MB floppy subsystems set productivity records for your IBM PC or compatible. Taking only minutes to install, we've got all the competition beat. TeamMate requires no special software or formatting diskettes. All you need is DOS 2.0!

We have a variety of configurations to fill a variety of needs. Like the TeamMate 1110 add-in 10MB Winchester, with power consumption so low it can be safely installed in even fully configured PC's. Or try the TeamMate 1103 KODAK 3.3MB floppy add-in subsystem. Also software transparent, it gives you 2.78MB of rugged floppy storage— perfect for Winchester backup or as a low cost alternative to Winchester disks. Or get the advantages of both with our TeamMate 1213 Winchester/Kodak floppy add-on combination. All at record prices.

Even TeamMate's customer support is fast. Should your subsystem need any service during the first 3 months of the 6-month warranty period, your dealer will be shipped a swapout unit within **24 hours!**

Ask for TeamMate IBM and Apple subsystems at your local computer dealer. Or call us at (408) 496-0434.

Team· Mate

TeamMate by Data Technology
2775 Northwestern Parkway
Santa Clara, CA 95051
TEL: (408) 496-0434
TWX: 910-338-2044

CIRCLE NO. 111 ON READER SERVICE CARD

that resides on the TIDY diskette. More detailed documentation is, luckily, unnecessary. TIDY can be invoked simply by entering:

TIDY source.tdy

The source.pas file is any Pascal source file, and the source.tdy file is a formatted source file produced by TIDY. The output file can be compiled immediately.

Programmers should be aware that TIDY allows the input and output file to be the same; if TIDY is invoked in this way, it writes the output file over the input file, destroying the original source code. This is inviting trouble, especially if the source code happens to contain syntax errors. Instead of formatting the code, TIDY might exit upon encountering the error, leaving the programmer with no code at all.

TIDY has a few bugs as well, but most of them are minor. For example, IBM Pascal allows the programmer to substitute square brackets for BEGIN and END. TIDY doesn't support that construction. It assumes that brackets are used only in array declarations, and if it finds them used in any other way, TIDY exits to DOS. The manual explains this restriction, and the people at Major Software have no plans to fix it. To handle the problem, TIDY would have to be almost as smart as the compiler itself.

Error-handling is also a slight problem. If for some reason TIDY is unable to handle the code it is fed, it exits to DOS and returns a run-time error message such as "stack underflow" or "stack overflow." It takes a while to interpret these messages to mean improper usage of brackets or too many levels of nesting. TIDY should handle these error conditions itself and return more meaningful error messages.

The people at Major Software are responsive when users point out problems with the program. When we reported some of the bugs we found, they fixed the problems (with the exception of the bracket problem) and sent us a new version of the program in two weeks.

Aside from the minor bugs, TIDY does its job well and quickly. Figure 1 shows a 51-line unformatted program, and Figure 2 shows the same program after it was run through TIDY. Formatting the entire program took only eight seconds (on a PC XT). We've run over 1,500,000 bytes of Microsoft Pascal source code through TIDY and are quite pleased with the results.

PASCAL UTILITIES

Like TIDY, PASCAL UTILITIES is inexpensive—$29.95. The package is made by Seven Valleys Software of Glen Rock, Pennsylvania, and

consists of four programs: three that perform different aspects of code formatting and one that produces cross-reference listings. These programs work with either DOS 1.1 or 2.0, can process code written in IBM Pascal, and require only 64K bytes of RAM. The cross-reference program works well, but is not within the purview of this article.

The three utilities are called PSTRIP, PRETTY, and PCASE, and each performs its own unique function. PRETTY handles the indenting of the text. It works much like TIDY does, with a few notable exceptions. First, it uses three spaces for each level of indentation; enough to cause a problem with "rightward creep." Second, PRETTY doesn't delete or adjust any of the spaces included in the code. Therefore, it doesn't adjust and standardize comments. Third, because PRETTY doesn't delete existing spaces, running the same program through PRETTY multiple times exaggerates the indentation each time. But that's where the second utility, PSTRIP, comes in.

PSTRIP strips off all leading spaces in the Pascal source code, making the code fit to run through PRETTY. Unless every line of the code being formatted begins at the left margin, PSTRIP should always be used before PRETTY.

In the default situation, PCASE converts all Pascal reserved words to upper-case letters and all variable names to lower-case letters. Another option does the reverse.

On the surface, PCASE seems more powerful than the equivalent feature in TIDY, but it is really less useful. TIDY ignores variable names, allowing programmers to establish their own conventions (possibly using first caps for certain variables or all caps for constants); PCASE arbitrarily converts variable names. Also, PCASE has a nasty bug that causes it to crash, without returning an error message, whenever the source code contains comments that begin with an exclamation point. This bug alone is enough to cause us to recommend against using PASCAL UTILITIES.

Like TIDY, PASCAL UTILITIES comes with only a few pages of documentation (a 17-page user manual). Unlike TIDY, however, some trial and error is necessary before users will be able to produce nicely formatted output with PASCAL UTILITIES.

To produce output similar to that produced by TIDY, programmers must first ensure that their code contains no comments that begin with exclamation points. After that is done, they can invoke the following commands:

PSTRIP source.pas source.pps
PCASE source.pps –o source.cas
PRETTY source.cas source.pty

The order of the commands is important. PSTRIP should be run first to strip off all the leading blanks, then PCASE should be run, and finally PRETTY can be run; it must be run last, because the capitalization conventions established by PCASE tell it how to indent.

In general, the formatting programs in the PASCAL UTILITIES package do not work well. Not only do they have problems formatting comments, but PSTRIP, PRETTY, and PCASE are slow—very slow when compared to TIDY. We tried the same sample program that we used with TIDY (except that we had to strip out all the comments with exclamation points). PSTRIP, PCASE, and PRETTY took almost five times as long (39 seconds) to format the program, and they do not provide all of the formatting features that TIDY provides.

The error-handling facilities of the PASCAL UTILITIES are even worse than TIDY's. When we tried to format a program that used square brackets for BEGIN and END, the PASCAL UTILITIES produced an output file, but the formatting was entirely incorrect. In other situations, such as when it encountered comments beginning with an exclamation point, the PASCAL UTILITIES would crash without explanation, requiring a reboot.

Although the formatting utilities have problems, programmers might want to buy the PASCAL UTILITIES just for the other program—the Pascal cross-reference program. It's one of few products of its kind around, and it works.

HOW THE OUTPUT LOOKS

Programmers have few options in telling a formatter how to format a program. Usually, the formatter produces code that looks one way, and one way only. So before buying a formatter make sure that its formatted output is acceptable.

Figure 2 shows the kind of output produced by TIDY; figure 3 shows output produced by the PASCAL UTILITIES package.

RECOMMENDATIONS

A Pascal formatter is more than just a luxury to serious Pascal programmers. It not only makes code more professional-looking but also helps programmers spot structural errors, makes it easier for them to maintain their code, and increases their overall productivity.

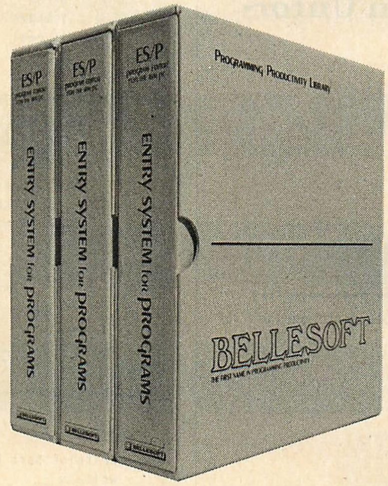
Of the products we've seen, we feel that TIDY is the one that most programmers will prefer.

TIDY
Major Software
66 Sylvian Way
Los Altos, CA 94022
415-941-1924
$39.00
CIRCLE 492 ON READER SERVICE CARD

PASCAL UTILITIES
Seven Valleys Software
P.O. Box 99
Glen Rock, PA 17327
717-235-5470
$29.95
CIRCLE 493 ON READER SERVICE CARD

BELLESOFT HAS CHANGED THE WAY PROGRAMMERS WORK:

PASCAL



C

COMPUTER AIDED PROGRAMMING

The Bellesoft Entry System for Programs (ES/P) has changed the way programmers will work in the future by putting computer aided programming (CAP) at their fingertips.

What word-processing programs did for writing text, Bellesoft has now done for generating programs: putting the computer to work to create a whole new way of writing software.

Bellesoft's ES/P speeds up software development by up to 50%, produces better programs and reduces debugging time. ES/P improves programming in five significant ways:

1. **On-Line Syntax Checking.** ES/P is more than a very smart editor: it automatically checks your program for correct syntax as you enter or change it.
2. **On-Line Formatting.** ES/P automatically provides a correct and consistent format, critical for program maintainability.
3. **Context-Sensitive Abbreviations.** ES/P builds program structures or skeletons with a single keystroke, provides correct keywords and punctuation, and saves up to half your typing time.
4. **Program Editing.** ES/P manipulates program structures correctly with a single keystroke, allows you to delete or move statements easily.
5. **On-Line Programming Help.** ES/P provides menus of correct program components and correct syntax as an aid to faster programming.

ES/P is now becoming available to computer manufacturers for inclusion in their software packages; to distributors and dealers for resale; to universities and schools as an aid to teaching programming; and to programmers who want to cut their work in half while writing better programs.

CIRCLE NO. 114 ON READER SERVICE CARD

To order, contact: Lifeboat Associates
1651 Third Avenue New York, NY 10028
(800) 847-7078

BELLESOFT

THE FIRST NAME IN PROGRAMMING PRODUCTIVITY

(206) 828-7282

Figure 1: MAJOR.SRC—An Unformatted Pascal Program

```
{ Pascal Sort Program }

program sort(input,output) ;
const
max_entries = 5;!length of sort table
type
table = array [1..max_entries] of real;
var
sort_table : table;!table of numbers
i : integer;!table index
{***************************************************************}
procedure bubble_sort(var number_table: table);!bubble sort routine

  {    This procedure sorts a table of numbers
  {    by using the tried and tested but not
  {    very fast bubble sort technique.     }

var
i:integer;!index
j:integer;!index
temp:real;!exchange variable
begin!bubble_sort
for i := 1 to max_entries - 1 do
begin
for j := i + 1 to max_entries do
begin
if number_table[i] > number_table[j] then!ascending order
begin
temp := number_table[i];!swap the table entries
number_table[i] := number_table[j];
number_table[j] := temp;
end;
end;
end;
end;!bubble_sort
{***************************************************************}
begin!main program
writeln('Sort Program');!print title
writeln;
for i := 1 to max_entries do!loop to read in numbers
begin
writeln('Please input a number') ;!prompt for a number
readln(sort_table[i]);!read the number
end;
bubble_sort(sort_table);!sort the table of numbers
writeln;
writeln('The list of numbers in ascending order is:');
for i := 1 to max_entries DO!loop to print all the numbers
write(sort_table[i]:2:4,'  ');!write each number
END.!main program
```

Figure 2: MAJOR.PAS—Listing 1's Program after Being Run through TIDY

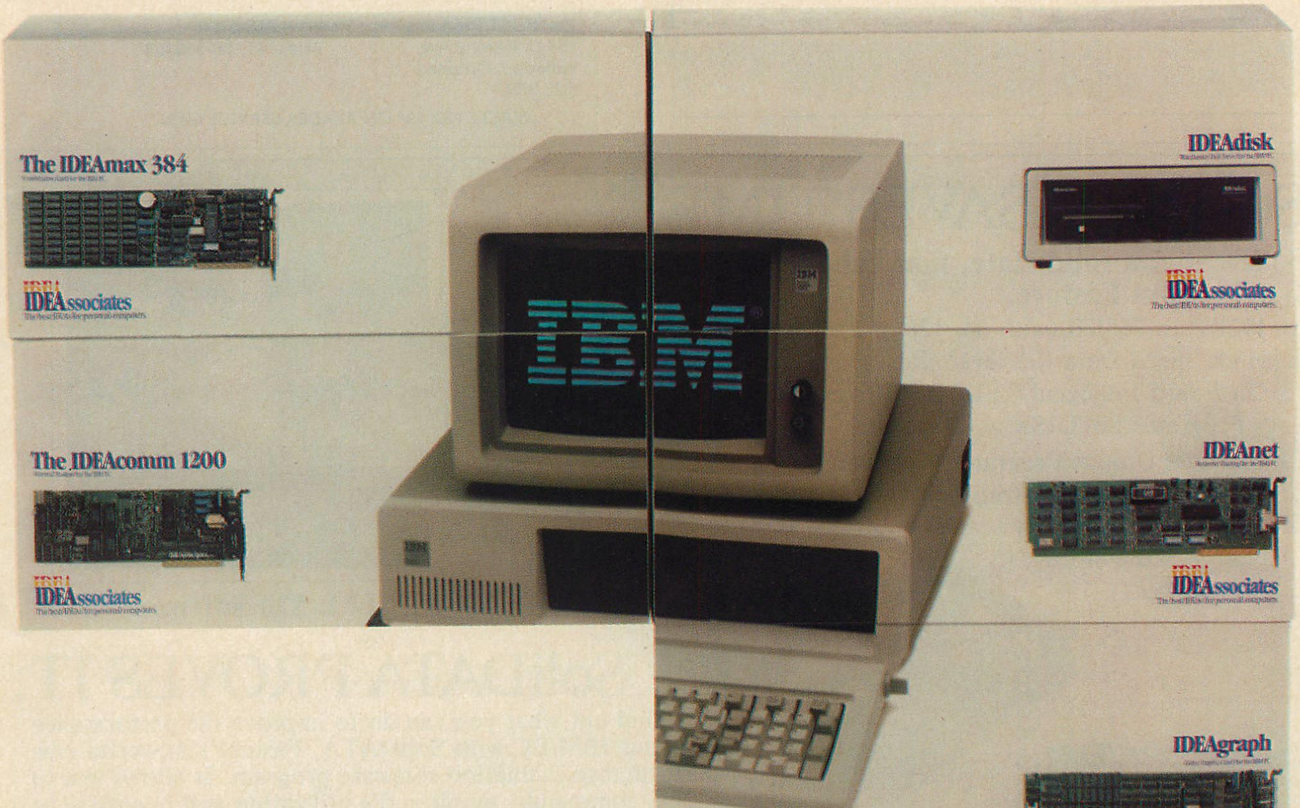
```
{ Pascal Sort Program }

PROGRAM sort(input,output) ;

  CONST
    max_entries = 5 ;                     !length of sort table

  TYPE
    table = ARRAY [1..max_entries] OF real ;

  VAR
    sort_table : table ;                  !table of numbers
    i : integer ;                         !table index
{***************************************************************}

  PROCEDURE bubble_sort(VAR number_table:table) ;!bubble sort routine
  {    This procedure sorts a table of numbers
  {    by using the tried and tested but not
  {    very fast bubble sort technique.     }

    VAR
      i : integer ;                         !index
      j : integer ;                         !index
      temp : real ;                         !exchange variable

    BEGIN                                   !bubble_sort
      FOR i := 1 TO max_entries - 1 DO
        BEGIN
          FOR j := i + 1 TO max_entries DO
            BEGIN
              IF number_table[i] > number_table[j] THEN !ascen. ord.
                BEGIN
                  temp := number_table[i] ;  !swap the table entries
                  number_table[i] := number_table[j] ;
                  number_table[j] := temp ;
                END ;
            END ;
        END ;
    END ;
                                            !bubble_sort
{***************************************************************}

  BEGIN                                     !main program
    writeln('Sort Program') ;               !print title
    writeln ;
    FOR i := 1 TO max_entries DO            !loop to read in numbers
      BEGIN
        writeln('Please input a number') ;  !prompt for a number
        readln(sort_table[i]) ;             !read the number
      END ;
    bubble_sort(sort_table) ;               !sort table of numbers
    writeln ;
    writeln('The list of numbers in ascending order is:') ;
    FOR i := 1 TO max_entries DO            !loop to print numbers
      write(sort_table[i]:2:4,'  ') ;       !write each number
  END.                                      !main program
```

Figure 3: SVC.PAS—Listing 1's Program (Minus Comments that Began with Exclamation Points) after Being Run through PASCAL UTILITIES

```
{ Pascal Sort Program }

PROGRAM sort(input,output) ;

CONST
      max_entries = 5;

TYPE
     table = ARRAY [1..max_entries] OF REAL;

VAR
    sort_table : table;
    i : INTEGER;
    {*******************************************************}

PROCEDURE bubble_sort(VAR number_table: table);

{    This procedure sorts a table of numbers
{    by using the tried and tested but not
{    very fast bubble sort technique.     }

VAR
    i: INTEGER;
    j: INTEGER;
    temp: REAL;
BEGIN
   FOR i := 1 TO max_entries - 1 DO
      BEGIN
         FOR j := i + 1 TO max_entries DO
            BEGIN
               IF number_table[i] > number_table[j]
                  THEN
                     BEGIN
                        temp := number_table[i];
```

```
                    number_table[i] := number_table[j];
                    number_table[j] := temp;
                 END;
           END;
       END;
END;
{**********************************************************}
BEGIN
   writeln('Sort Program');
   writeln;
   FOR i := 1 TO max_entries DO
      BEGIN
         writeln('Please input a number') ;
         readln(sort_table[i]);
      END;
   bubble_sort(sort_table);
   writeln;
   writeln('The list of numbers in ascending order is:');
   FOR i := 1 TO max_entries DO

      write(sort_table[i]:2:4,'  ');
END.
```

We have a missing piece for your IBM PC.



Forget IRMA. Here's IDEAcomm 3278. The best idea in mainframe communications.

The IDEAcomm 3278 gives you all the functionality of your IBM PC, plus direct access to mainframe information and applications. Without having to deal with two terminals.

But how does IDEAcomm 3278 stack up against IRMA? Well, you know IRMA has a reputation for being fast. But the IDEAcomm 3278's powerful file transfer moves data faster. Up to 60% faster.* So response time is nearly instantaneous.

Our file transfer is real file transfer. You can spool files from the mainframe to the PC and back again by hitting just one key. One key to send a file, one key to receive a file. Not IRMA. She sends you jumping through hoops to use her file transfer.

IRMA also has a reputation for being flexible. IDEAcomm 3278 is more flexible. You can specify how to change the 3278's programmable keys to best suit your applications. And our 3279 emulation mode lets you configure colors any way you want. Again, the IDEAcomm 3278 lets the user decide. You'll find IRMA likes to make her own decisions.

The IDEAcomm 3278 makes life easier for your data processing department, too. Included in our reference manual is a programmers' section which outlines how to automate functions which previously had to be performed manually.

Are you getting the idea that the IDEAcomm 3278 is a lot more than a plug-in card? You're right. Other features include:

- 100% compatible with Micro Tempus and Paramin mainframe link packages.
- IDEAcomm 3278 interfaces to IBM 3274, 3276 controllers.
- Includes TSO and VM/CMS file transfer.
- File transfer sends all text files to the mainframe. HEX files can be sent to the host, then spooled to other users.

Best of all, IDEAcomm 3278 costs less. Only **$995** including software. There are no hidden charges.

Ask for IDEAcomm 3278 and the entire IDEA family of products at all Computer Depot, Math Box, Amerisource, and Northeast Computer stores, plus selected Computerland, Inacomp, and MicroAge® locations. For the name of the IDEA store nearest you, call us at **800-257-5027**.

See why the best IDEA in mainframe communications is the piece you've been missing for your PC.

*In benchmark studies, IDEAcomm 3278 transmission speed on a local mode 3274 was up to 60% faster than IRMA.

CIRCLE NO. 136 ON READER SERVICE CARD

IDEAssociates

The best IDEAs for personal computers.

IDEAssociates, Inc., 7 Oak Park Drive, Bedford, MA 01730 (617) 275-4430 Telex 94-8245. IDEAssociates S.A., 5, Route de Chêne, CH1207 Geneva, Switzerland (22)86.11.19 Telex 428 388 KBSCH

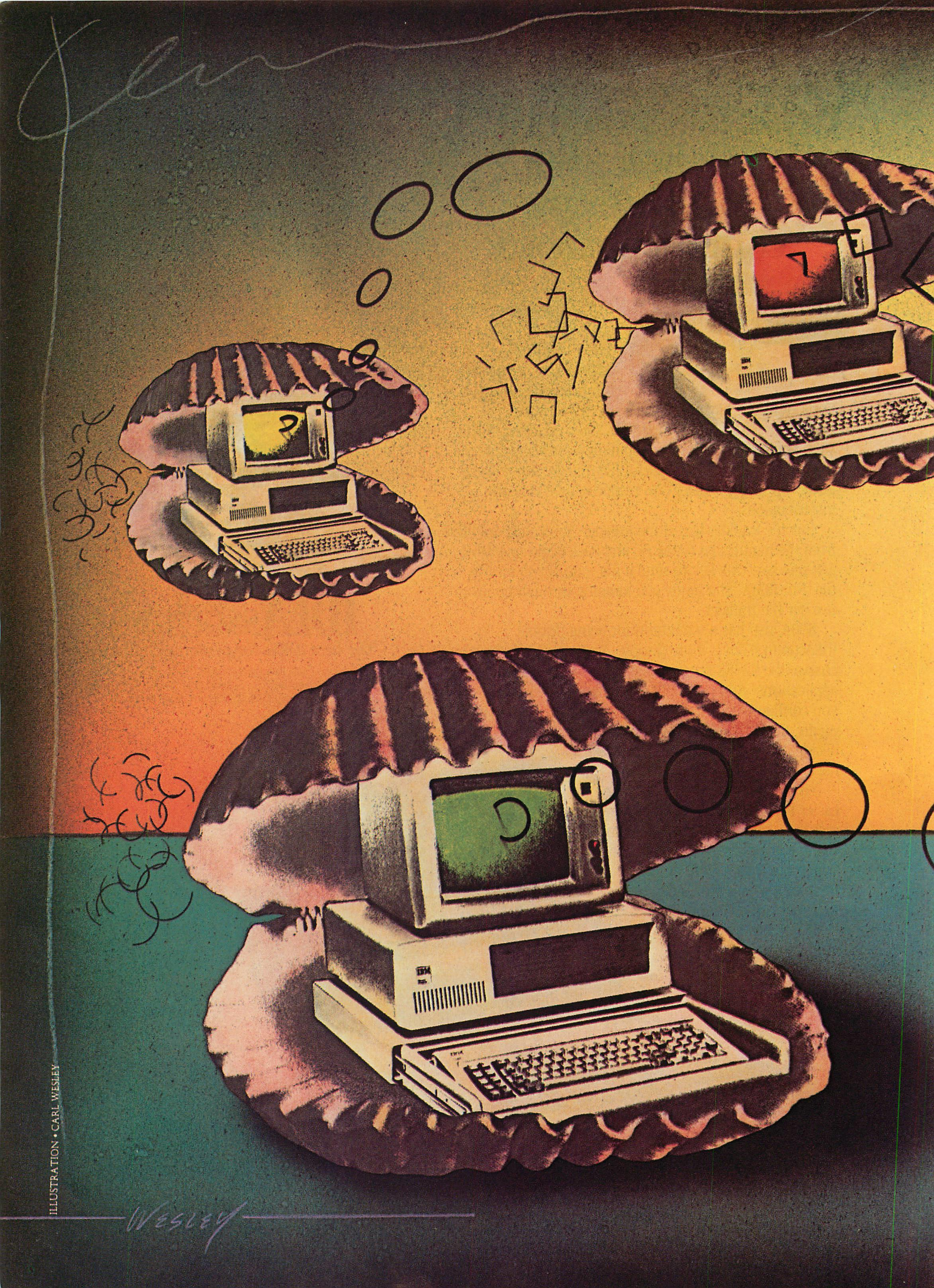


ILLUSTRATION • CARL WESLEY

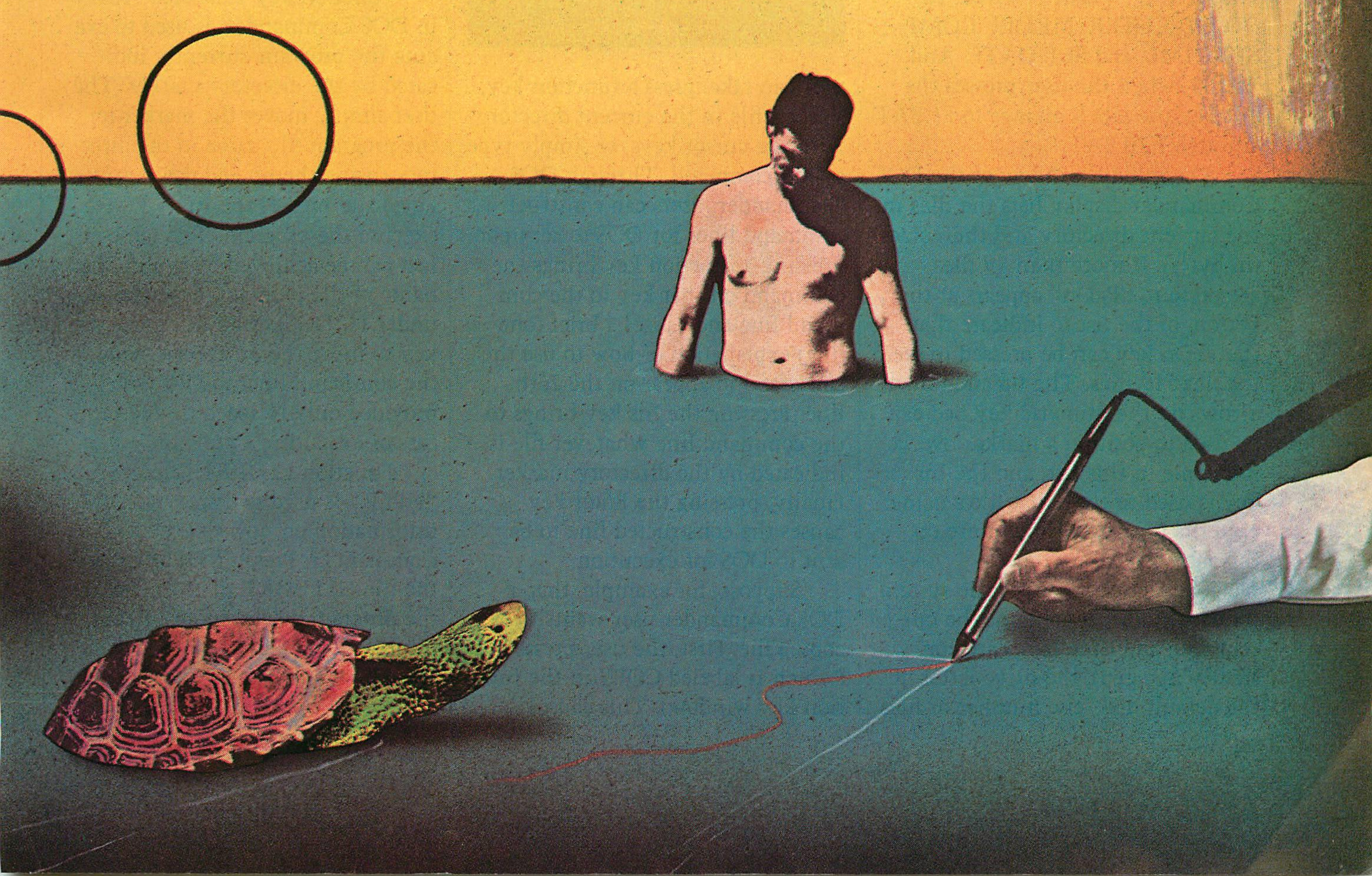
DOS SHELLS
CONTINUED

ARTHUR A. GLECKLER

DOS Commander and DOS Ease succeed in making PC-DOS somewhat more enjoyable to use; DOS Shell does not, although it can help organize the XT's fixed disk.

PC-DOS is a well-thought-out, practical operating system, but it is certainly neither easy to use nor friendly. Novice users often find it difficult to understand, much less to use effectively. Even experienced computer users find PC-DOS's syntax and command structure confusing and, at times, frustrating.

Enter shells. Shells are programs that "surround" the disk-operating system and provide their own syntax and methods of operation, different from the original DOS methods. By hiding the details and intricate syntax of DOS, a good shell reduces confusion and increases the efficiency of the user. In this article three shells for PC-DOS are reviewed: DOS Commander, DOS Ease, and DOS Shell.



DOS COMMANDER

DOS Commander, from Applitek, Inc., works only with PC-DOS 2.0. The package includes an instruction manual and two versions of the program, one for color displays and one for monochrome displays. Both versions work exactly alike, except that the color version makes use of the IBM Color Graphics Adapter to provide a more attractive display.

When DOS Commander is invoked, it sets up a full-screen display that is divided into three windows and a command area (see photo 1). On the left side of the screen is a diagram that represents the arrangement of the function keys on the PC keyboard. A description is placed over each key.

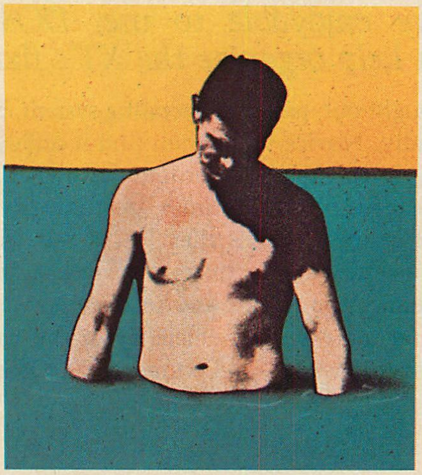
DOS Commander uses the Shift and Alt keys in combination with the function keys to increase the number of available functions to 30. Normally, the functions are labeled RUN, COPY, TYPE, FORMAT, ERASE, RENAME, PRINT, MODE, DIR, and HELP. When the Shift key is pressed, the functions change (on the display) to DISKCOPY, DISKCOMP, VERIFY, TREE, RESTORE, CHDIR, MKDIR, COMP, SET TIME, and SET DATE. And finally, when the user presses the Alt key the labels change to F1.BAT through F.10.BAT.

The middle window of the DOS Commander display lists the files in the current directory and their sizes (in bytes). If more than 19 files are present, "PgDn" appears at the bottom of the list to indicate that the PgDn key can be pressed to see the next 19 files. The up and down arrows on the numeric key pad can be used to position a marker over a file name in this list, and the Ins key is used to copy that file's name onto the command line (more on this later in the article).

The right window is the system status window. It contains the current date and time (both are constantly updated), the drive letter of the default disk, the number of files in the current directory, the number of bytes free on the disk, the volume ID of the diskette, amount of total memory (both installed and free), and the status of the Caps Lock and Num Lock keys.

Under the three windows is the command area. This area consists of three lines of text. On the first line the name of the current directory is displayed. User commands are typed on the second line (after the traditional "A>" prompt), and the third line always contains a brief message explaining what to do next.

There are three basic types of actions that the DOS Commander



user can take: press a function key, select a file in the current directory with the cursor keys, or simply type a DOS command. Through DOS Commander, users can construct command lines for DOS to act upon. Pressing a function key brings the command on that key to the command line and causes a brief (one-line) explanation of how to use the command to appear on the 25th line. Pressing the Ins key brings to the command line whatever file is indicated by the directory marker. Finally, pressing the Enter key causes the constructed line to be sent to DOS for execution.

Suppose, for example, that a DOS Commander user wants to copy a file. First, the user presses F2 (which is labeled COPY in the function key window). This causes "COPY" to appear on the command line, and the message "Copy a file: Enter 'from' filename followed by 'to' filename" to be displayed. Next, the user types the name of the file to copy from (including wildcards, etc., because DOS Commander places no restrictions on program arguments). This name is followed by a space and then the name of the file to copy to (again, including wildcards if desired).

Once the command line has been entered, the user presses the Enter key to have the entire line executed. DOS Commander simplifies the process by giving help messages, but it does not eliminate the need to understand PC-DOS syntax; all program parameters are still entered in standard PC-DOS form.

Although DOS Commander is designed for construction of command lines that are to be sent to DOS, it does not provide facilities for editing these lines (other than by using the backspace key). This is an unfortunate omission, because the only way a mistake can be corrected is by retyping a line.

Function key 1, labeled "RUN" in DOS Commander, is used to execute the program currently indicated by the directory marker. The user simply moves the marker to the program file name in the directory (using the up and down arrow keys) and presses the RUN key to execute the program. The same effect can be achieved by typing the name of the program (exactly as under PC-DOS) or by using the Ins key to bring the program's name to the command line; the RUN key is provided only to speed up the process of executing a program.

Function key 10, labeled "HELP," is used to display general information on how to use DOS Commander. Pressing HELP causes the file HELP.BAT to be executed; the program displays four screens of information, pausing on each screen to wait for the user to press a key. HELP.BAT and HELP1.TXT

Discover Solutions With Vision

Dealers are finding Pacific Datanet's Vision™ hard disk sub-systems are today's personal computer users' choice for mass storage sub-systems.

The Vision Series is more than illusionary—read on!

Capacity A variety of sub-systems—12/10Mb, 19/17Mb, 26/21Mb, 40/32Mb, 50/43Mb, 70/62Mb, 105/92Mb, and 140/125Mb.

Software Menu-driven software makes installation simple with the IBM PC, XT, or IBM compatible.

Complete Sub-System Vision Series are complete. Sub-systems include a 5¼" Winchester, our controller, host adapter, heavy duty international power supply, software and cables.

Fast Tape Backup 5¼," ¼" tape streamer backs up at a consistent 5Mb per minute, using standard 90 ips, ¼" tapes.

Error Correction Sub-systems provide 8 bit error correction.

Reliability Stringent quality control assures reliable operation.

Price Prices start at $2,195 for the hard disk sub-system, and $1,995 for the streaming tape backup.

Dealer Inquiries Invited
PACIFIC DATANET, LTD.
4701 Patrick Henry Drive, Bldg. 9
Santa Clara, CA 95054
(408) 980-0693 Telex: 759341

**VISION is a trademark of Pacific Datanet, Ltd.*
**IBM is a trademark of International Business Machines Corp.*
CIRCLE NO. 169 ON READER SERVICE CARD

Pacific Datanet

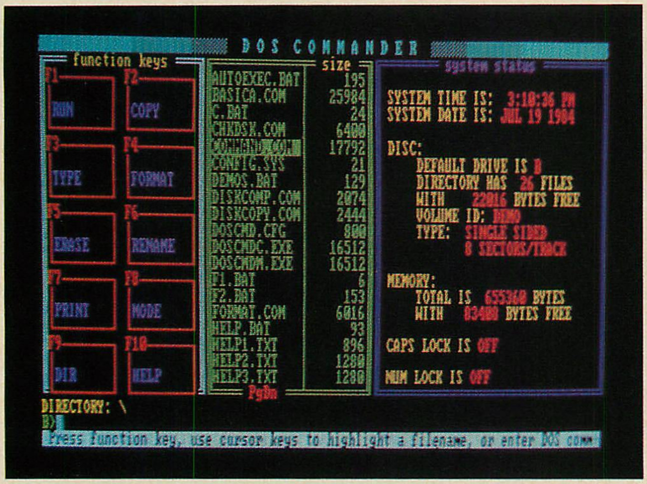


Photo 1: DOS Commander Status Screen

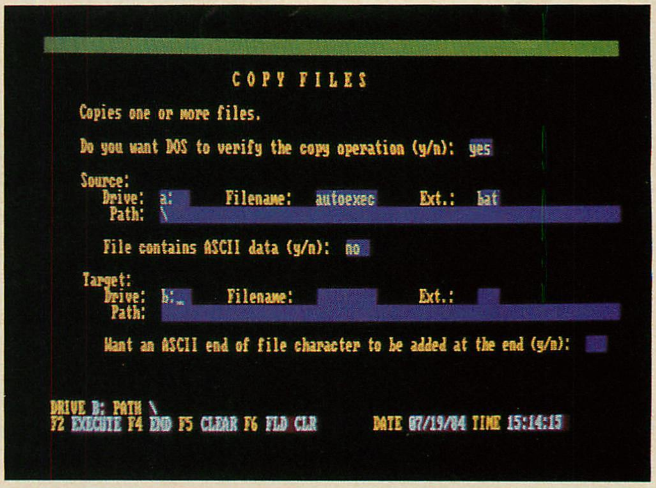


Photo 2: DOS Ease "COPY FILES" Function Screen

through HELP4.TXT, which contain all of the help information, must be in the current directory in order for HELP to work.

DOS Commander's authors have provided an easy method to extend the program to suit individual users' needs. If the Alt key is held down while a function key is pressed, one of 10 batch (.BAT) files will be executed. These files, named F1.BAT through F10.BAT, can contain batch programs to handle operations that are used often. For example, if a popular integrated business program is used frequently by a DOS Commander owner, the file F1.BAT could contain

C:DOITALL /WORD /CALC /COMM /BASE

making the process of starting the program quicker and easier. Note that the batch files must be in the current directory for this to work.

The DOS Commander manual specifically states:

> DOS COMMANDER also requires the presence of a special file called COMMAND.COM to be [sic] on the disc from which COMMANDER is initially started, typically Drive A. Even if you direct DOS COMMANDER to Drive B (by typing B:) COMMAND.COM needs to be on the disc that is in Drive A.

This is not the full story. While testing DOS Commander, I found that the program will not operate unless COMMAND.COM is in the root directory of the default drive at all times. Even after DOS Commander finds COMMAND.COM on the default drive, however, it looks for COMMAND.COM on the boot drive (A for floppy systems, C for XTs). Thus, contrary to what the manual says, COMMAND.COM must be both in the root directory of the default drive *and* in the root directory of the boot drive.

DOS Commander also requires that all programs to be executed—including DOS utilities such as FORMAT and MODE—be in the current directory on the default drive. If the FORMAT function key is pressed and FORMAT.COM is not in the current directory, DOS (not DOS Commander) will return the "Bad command or file name" message. This will also happen if the user tries to invoke any DOS utility not in the current directory. This defeats the purpose of DOS Commander, which is to make the operating system easier to use.

A reasonable solution to this problem would be for Applitek to change DOS Commander so that it would search for DOS utilities that appear on the function keys in a predesignated drive and directory (that is, in the root directory or on drive A). This improvement would allow the user to change directories without losing the DOS utilities.

DOS Commander users can take matters into their own hands and set up directory search orders using the PATH command from DOS. This is not an ideal solution, however, for two reasons: first, most novice users (the primary users of DOS Commander) probably will not know how to set up a PATH. Second, considerable time can be spent waiting for DOS to find a program that could be in any directory in the search path, as well as in the default directory.

Although the DOS Commander system-status window displays the volume ID of the default disk, it does not do this with reliability. When presented with several disks that had volume IDs accepted by PC-DOS, DOS Commander displayed either garbage or a "(none)" message. This is not a major hindrance, but it is a problem.

The DOS Commander documentation is very short (18 pages) but complete. Except for its discussion of the problem with COMMAND.COM mentioned earlier, the manual is accurate and easy to understand, although its writing style leaves something to be desired.

The manual gives an accurate and honest summary of DOS Commander's usefulness:

> Although DOS COMMANDER does provide prompts for DOS commands invoked by the function keys


"My computer keeps track of 135,000 cubic feet of wholesale auto parts."
The more work you do, the more you need to store. And the harder it is to work with stacks of floppy diskettes.
"My Sysgen hard disk stores the whole lot and runs a whole lot faster."
Sysgen-II-G™ hard disk systems store everything—up to 20 Megabytes—conveniently. You can look at any program or file in seconds. But . . .
"And this lets me sleep at night."
Any hard disk can fail—and lose your information. So every Sysgen-II-G hard disk includes ¼″ cartridge tape back-up—up to 20 Megabytes per cartridge. (We make back-up systems for other hard disks, too.) So you can protect your files in minutes. Instead of spending hours re-entering them.
Sysgen.
Because a hard disk without tape just doesn't make sense.
Sysgen products for the IBM® PC, PC XT™, and other personal computers: Economical, 10- and 20-Megabyte hard disk systems with tape back-up. Or 10-Megabyte tape back-up for the IBM PC XT.
Go to your local computer dealer. Ask for a demonstration of Sysgen back-up systems. And find out how to make sense of your storage.
47853 Warm Springs Blvd., Fremont, CA 94539
(415) 490-6770 Telex 4990843
SYSGEN
INCORPORATED
IBM PC-XT and IBM PC are trademarks of International Business Machines Corporation. SYSGEN-II-G and IMAGE are trademarks of Sysgen, Incorporated.

to remind the user of the correct syntax, this is not considered to be a teaching system. At least a rudimentary knowledge of DOS functions and access to the DOS Users manual is suggested.

Although the novice computer user could be just as confused by DOS Commander as by PC-DOS itself, DOS Commander is helpful for those who understand the fundamentals of using PC-DOS but do not have extensive experience using it.

Unfortunately, in some cases, DOS Commander requires a better understanding of PC-DOS than does PC-DOS itself. For example, a DOS Commander user might be confused by the fact that DOS utility programs become unavailable when not in the current directory, even when the labels on the function keys seem to indicate that the utilities are still available. The nature of the program is such that DOS utilities look like integral parts of DOS Commander, not like separate programs.

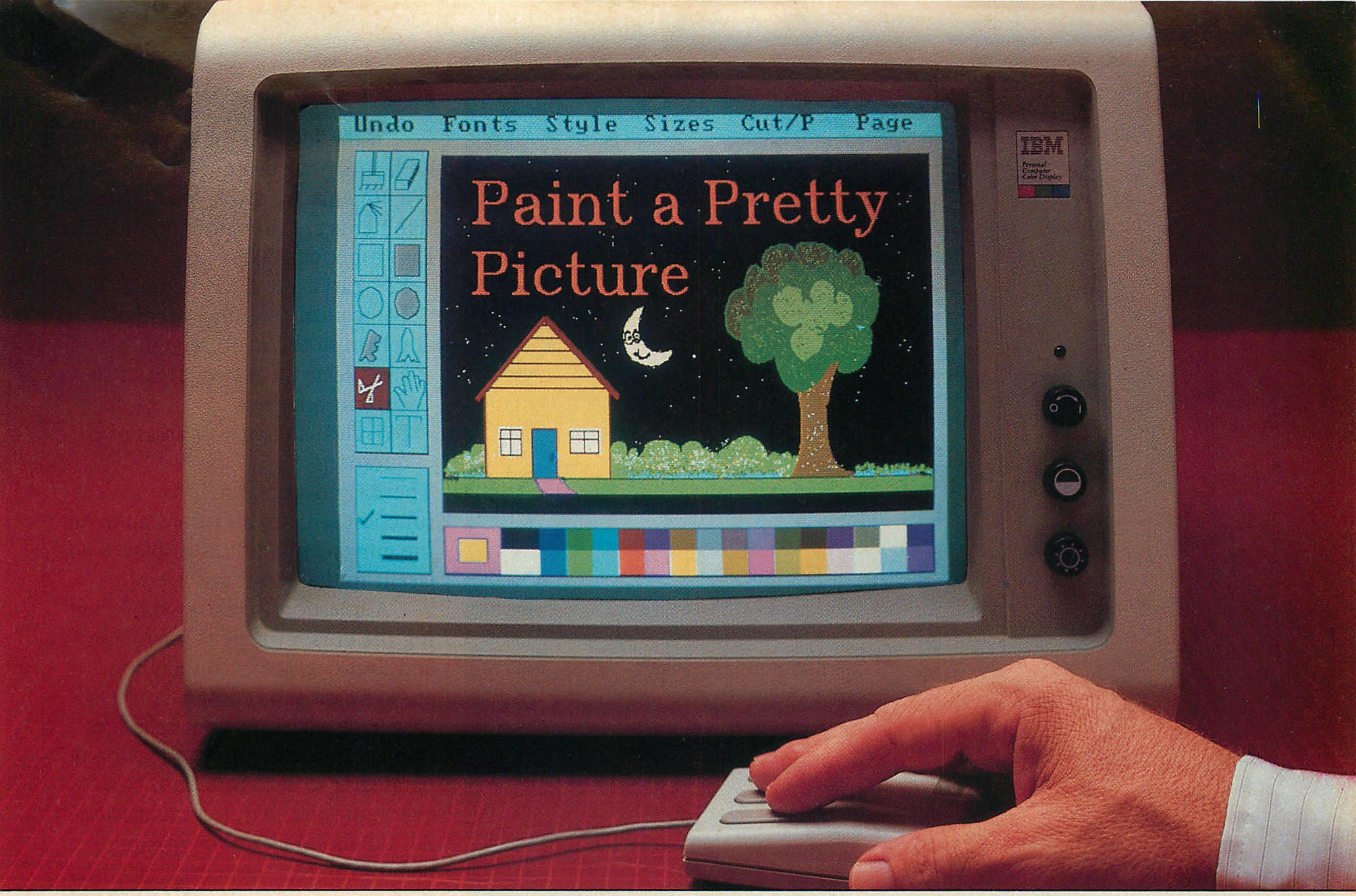
Overall, DOS Commander is a well-designed program that makes PC-DOS a friendlier environment in which to work. Its screen display looks good and gives a lot of information at a glance. In addition, the explanations of DOS functions that DOS Commander provides are simple and to the point, and they provide the extra hints needed to make PC-DOS usable by novices.

DOS EASE

Software Solutions, Inc.'s DOS Ease, like DOS Commander, is used to compose command lines to be sent to DOS. DOS Ease, however, goes further toward eliminating the need to understand complicated DOS syntax. The user selects a function from a menu of 32 choices and responds to questions about what the computer should do. DOS Ease constructs a PC-DOS 2.0 command line to carry out the operation and sends it to DOS for execution.

The main screen of DOS Ease is divided into six sections, each dealing with a different type of DOS function: FILE, DISK, DIRECTORY, SYSTEM, FIXED DISK, and OTHER. Each section contains brief, numbered descriptions of different functions. The lower right-hand corner of the screen displays the date and time continuously, and the lower left-hand corner shows function key assignments (F1 displays one page of help information, F4 exits DOS Ease, and F6 cancels the current function).

When the main screen is displayed, the user enters the two-digit number that corresponds to the function that he wants. DOS Ease immediately displays a screen full of questions and helpful information concerning that particular function. The questions are presented as a form to be filled out by the user, with a blank for an answer located next to each question.



You've seen the Macintosh, now see PC Paintbrush

PC Paintbrush

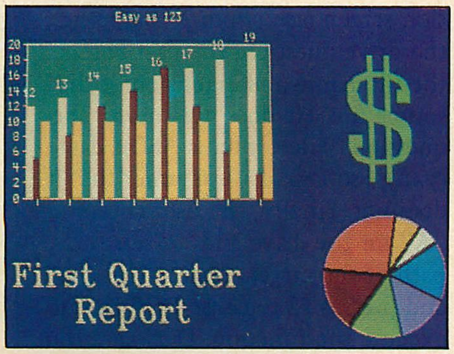
After many months of testing, ZSOFT Corporation is pleased to announce the versatile, colorful, PC Paintbrush.

Not just 2 colors, but 4, 16, or even 256 colors. Not just 320x200 resolution, but 640x200, 640x400, 720x350, 720x704, or more (depending on display adapter).

PC Paintbrush is available NOW for the IBM PC and compatibles, with 192K of memory and DOS 2.0 or above. We support an astonishing variety of hardware; check the lists below. Soon: HP150 and Corona 640x325 versions.

For Business, too

Besides being lots of just plain fun, we've designed PC Paintbrush for serious business. Our character sets (all 6 of them) are the best looking in the industry. Our printout is second to none.



Want to improve humdrum bar graphs? With PC Paintbrush, you can grab a chart or graph from another program (such as Lotus' 1-2-3 or Soreim's SuperCalc3), then embellish it with PC Paintbrush. Take a plain 2 or 4 color bar graph and produce a stunning 16 color slide or printout (requires appropriate display adapter).

We offer special group licensing agreements, as well as inexpensive customization services for your business' special requirements.

Astonishing Resolution

No need to buy additional hardware. We've made sure that PC Paintbrush will work with your printer, your display, and your application.

You can print on B/W printers from IBM, Epson, NEC, Okidata, C-Itoh, and Gemini; or color printers from Diablo, Tektronix, PrintaColor, Hewlett Packard (HP7475 Plotter), Quadram, and Transtar. With the color printers, pick from 25 different colors. With black and white printers, pick from 16 different shading patterns. Color printers never looked so good.

The display possibilities with PC Paintbrush are equally amazing. We COMPLETELY support display cards by Amdek, Quadram, Tecmar, Hercules, IBM, Mindset, Scion, and even the PCjr 16 color display. Just imagine 640x480 resolution in 16 brilliant colors (that's 9 times the resolution of the IBM color card!), or 720x704 resolution in 3 shades of grey.

Blinding Speed

You hate waiting and so do we. All of our graphics are written entirely in assembler, for the fastest possible results.

We provide Products, not Promises

No promises, just fact: we are currently shipping PC Paintbrush.

All ZSOFT software comes with a real warranty, and our service is second to none. Just ask some of our customers (such as IBM, Sohio, Michelin, Salomon Bros., Sperry, Westinghouse, U. of Pittsburgh . . . shall we go on?).

For those interested in updates, a single small fee entitles you to updates for one year; no catch: no additional costs, no need to return diskettes, no postage fees.

Schools

Schools and universities please call for educational discounts.

Z·SOFT

370 HERMITAGE CT. SW
MARIETTA, GEORGIA 30064 • 404-425-3766

CIRCLE NO. 203 ON READER SERVICE CARD

In the lower left-hand corner of each function screen, function key assignments are displayed. These assignments are the same for all DOS Ease functions. They are:

F2—carry out the function;
F4—cancel the function and return to the main screen;
F5—clear the answer area of the current question;
F6—clear the answer areas of all questions.

There is no special help key on the function screens; all information on the current function is displayed adjacent to the questions.

A question is answered by moving the cursor (using the arrow keys and Enter) to the corresponding answer area and typing an answer. DOS Ease asks every possible question about each function, leaving no parameter unaccounted for. Not every question requires an answer. All unanswered yes/no questions default to "No" responses. However, if the user presses the F2 EXECUTE key without answering an essential question (such as the name of the file for the RENAME command), DOS Ease will respond with the message, "This field must be filled"; the program will then position the cursor over the neglected question and wait for a valid answer.

The process of copying the DOS Ease program from the current directory to the root directory in drive A: can serve as an example of DOS Ease operation. From the main screen, the user selects the "Copy files" choice in the FILE section by typing "12." The main screen is replaced by the "COPY FILES" function screen (see photo 2). Skipping the drive question, the user enters the file name and extension of the source file. He does not answer the question "File contains ASCII data (y/n)," so DOS Ease assumes that it does not. Finally, the user enters "A" for the drive and "/" for the path of the target (destination) file, and presses F2 to cause DOS Ease to execute the function. DOS Ease con-

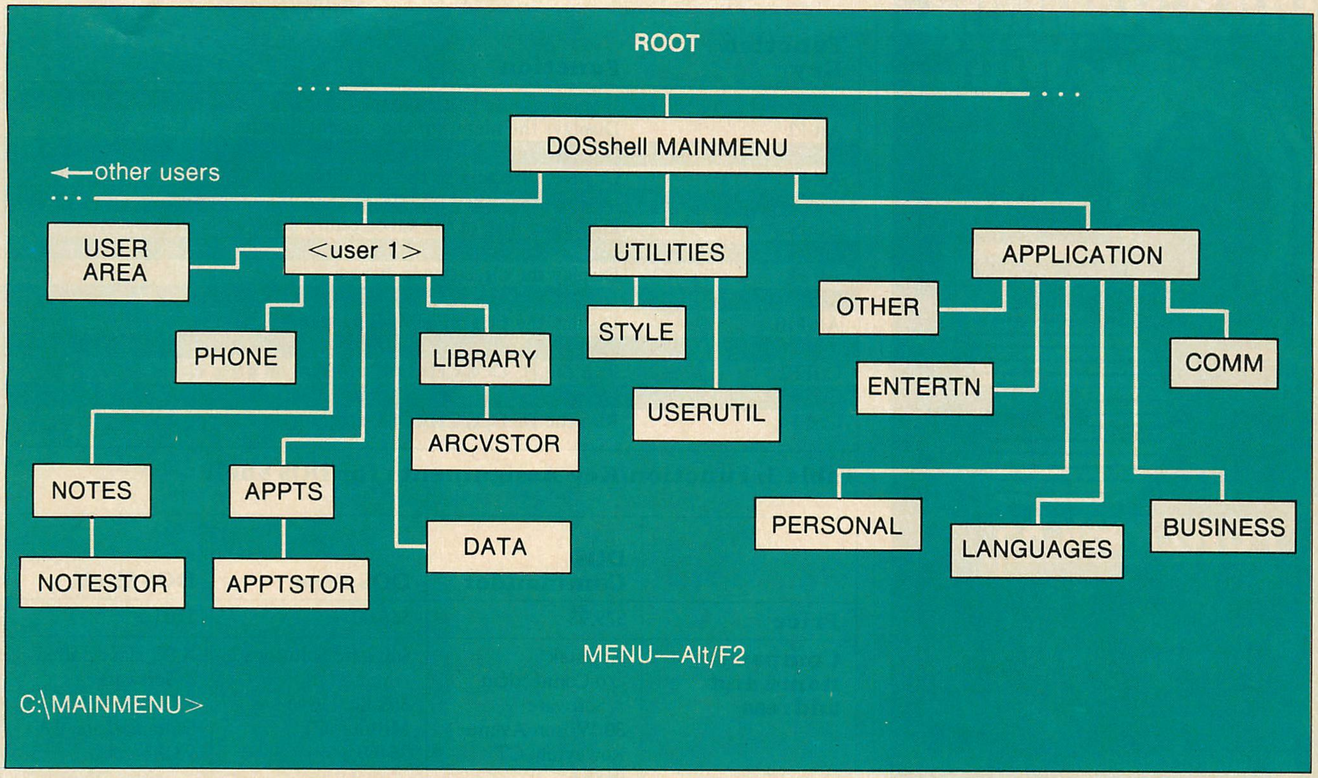


Figure 1: DOS Shell Subdirectory Map

structs the following command line and sends it to DOS:

COPY dosease.exe /B a:* /B

DOS Ease, like DOS Commander, requires that DOS utilities (FORMAT, CHKDSK, etc.) be in the default directory whenever they are used or be accessible via a directory search order defined through the PATH command. DOS Ease requires only one copy of COMMAND.COM to operate; this should be present in the root directory of the boot disk.

The DOS Ease manual is well written and complete. It clearly explains the purpose of the program and how to install and use it. The documentation includes a glossary of PC-DOS terminology and a map of DOS Ease keyboard layout.

DOS Ease makes it almost impossible for an illegal command syntax to be sent to DOS. It ensures that all essential questions are answered, checks the user's answers, and explains all parameters clearly and simply. In addition, DOS Ease can carry out virtually any function that DOS can (from changing the date to setting the status of BREAK). DOS Ease considerably eases the process of interacting with PC-DOS. Although it does not speed up DOS use for experienced users as DOS Commander can, DOS Ease does an excellent job of isolating the user from the complexity of PC-DOS.

DOS SHELL

DOS Shell, from XTC Handcrafted Software, is different from the shells discussed above. It is designed for use only with fixed-disk systems (that is, XTs or PCs with the expansion chassis or compatible fixed-disk drives) under PC-DOS 2.0. DOS Shell makes no attempt to isolate its user from PC-DOS command-line syntax; rather, it enhances PC-DOS by organizing the fixed disk into subdirectories and "user areas" and making it easy to move between them. DOS Shell is not a program as such; it is a collection of batch files and one BASIC program that makes a fixed disk easier to use.

DOS Shell can be installed on a fixed disk, whether or not the disk is empty. During installation, it creates subdirectories and places help files and batch programs in each subdirectory (figure 1 shows the structure of the subdirectories).

To use DOS Shell, the user must insert the command

device = ANSI.SYS

into the CONFIG.SYS file. One use for the ANSI.SYS program is to allow programs to redefine the function of any key on the keyboard (see "Defining Function Keys Using ANSI.SYS," PC Tech Journal, March 1984, page 77). DOS Shell makes use of ANSI.SYS to change the assignments of the function keys, as shown in table 1.

Using DOS Shell involves pressing function key combinations to arrive at a subdirectory, then executing a batch file. The batch files in each subdirectory are 1.BAT, 2.BAT, etc., so that the user must press only a numeric key and Enter in order to activate a function.

Function Key	Function
Alt-F1	Displays the help file for the current directory
Alt-F2	Displays the menu for the current directory
Alt-F3	Changes directory to the DOS Shell main menu
Alt-F4	Changes directory to the root directory
Alt-F5	Displays these key assignments
Alt-F6	Displays a map of the subdirectory structure
Alt-F7	Display the directory of drive A
Alt-F8	Display the directory of drive C
Alt-F9	Display the files on drive A by date
Alt-F10	Display the files on drive C by date
Ctrl-F1	Quit (prepare to turn off the system)
Ctrl-F2	Run the PC-DOS CHKDSK program on drive C
Ctrl-F3	Format a floppy disk
Ctrl-F4	Run the PC-DOS CHKDSK program on a floppy disk

Table 1: Function Key Assignments for DOS Shell

	DOS Commander	DOS Ease	DOS Shell
Price	$79.95	$66.00	$70.00
Company name and address	Applitek c/o Connecticut Software 30 Wilson Avenue Rowayton, CT 06853	Software Solutions, Inc. 305 Bic Drive Milford, CT 06460	XTC Handcrafted Software P.O. Box 902 Santa Barbara, CA 93102
Memory space used when installed	16624	68624 (Includes second copy of PC-DOS invoked by DOS Ease)	0 Not memory-resident
Brief description	Speeds up PC-DOS but does not make using DOS easier	Well-thought-out compromise between PC-DOS-like operation and ease of use	Helps keep a fixed disk organized but does not make DOS easier to use
Disk space required when installed	0 (Memory resident)	50944	180935

Table 2: Summary of Features for the Three Shells

DOS Shell divides the fixed disk into three main areas: the applications program area, the utility area, and the user area. The applications program area is, in turn, divided into business, communications, entertainment, languages, personal, and other subdirectories. Each of these subdirectories has its own file of help information. The authors of DOS Shell intend for each user to install applications programs into the appropriate subdirectory.

The second major division of the fixed disk, the utility area, handles two functions: setting the print mode of an Epson or IBM Matrix printer and setting up subdirectories for new users of the system.

The primary purpose of DOS Shell is to help organize the fixed disk. This is done by dividing the disk into the three areas mentioned above and by providing separate subdirectories within the user area for each person who uses the computer system. To add a new user area, the DOS Shell user moves to the utilities subdirectory and activates the "Add a new user" function. DOS Shell asks for the name of the new user and creates a new user area accordingly. The next time this user "logs onto" the system, he or

she will have to type "LOGON name." This tells DOS Shell which subdirectories to use for that person's DOS Shell applications programs (again, see figure 1).

In each user area, DOS Shell provides a program to keep track of random notes, a telephone directory, an appointments calendar, and a database program. All except the database program are written entirely using batch files. Unfortunately, all of the programs are awkward to use and not satisfactory. Although DOS Shell shines in organizing the fixed disk, its applications programs leave a great deal to be desired.

For example the "Notes and Memos" subdirectory lists the following functions:

List notes
Write notes
Read notes
Print notes
Delete notes
Go to user menu.

Each of these functions is implemented in a batch file (1.BAT through 6.BAT). When the "Write notes" function is invoked, DOS Shell displays a message describing the procedure for storing a note:

```
ALL NOTES SHOULD CONTAIN
AS THEIR FIRST LINE THE
WORD "SUBJECT," FOLLOWED
BY A SHORT DESCRIPTION.
THIS IS USED BY SELECTION #1,
WHEN LISTING NOTES.
  COPY CON: TEST.NTS
  SUBJECT this is a test note
        .
        .
        .
  F6 <return>.
```

Clearly, this is not an ideal program; in fact, it is not a program at all. All the other DOS Shell programs included with DOS Shell use similar techniques and are full of nasty quirks and inconveniences. This is largely because the authors of DOS Shell chose to implement all their programs (except the database program, which is written in BASIC

but is similarly hard to use) as batch files; PC-DOS 2.0 batch files simply cannot do what is needed to make these programs viable.

The DOS Shell manual is well written and explains the installation and use of DOS Shell in full detail. DOS Shell serves well as a method of organizing a fixed disk, but it fails miserably in its user applications programs. Nevertheless, it can make using a fixed disk easier, especially in organizations in which more than one person will be using the same computer at different times.

RECOMMENDATIONS

Table 2 summarizes the characteristics of the three shells reviewed here. DOS Ease, although closer to PC-DOS than might be desirable, virtually eliminates problems with DOS syntax. The user has only to decide what options to choose; DOS Ease constructs a valid DOS command line from this information. Because DOS Ease is able to do just about anything that PC-DOS can, without complicated syntax, it is an excellent program for anyone who is unfamiliar with PC-DOS. However, DOS Ease, with its forms-filling type of interaction, will slow the more experienced PC-DOS user.

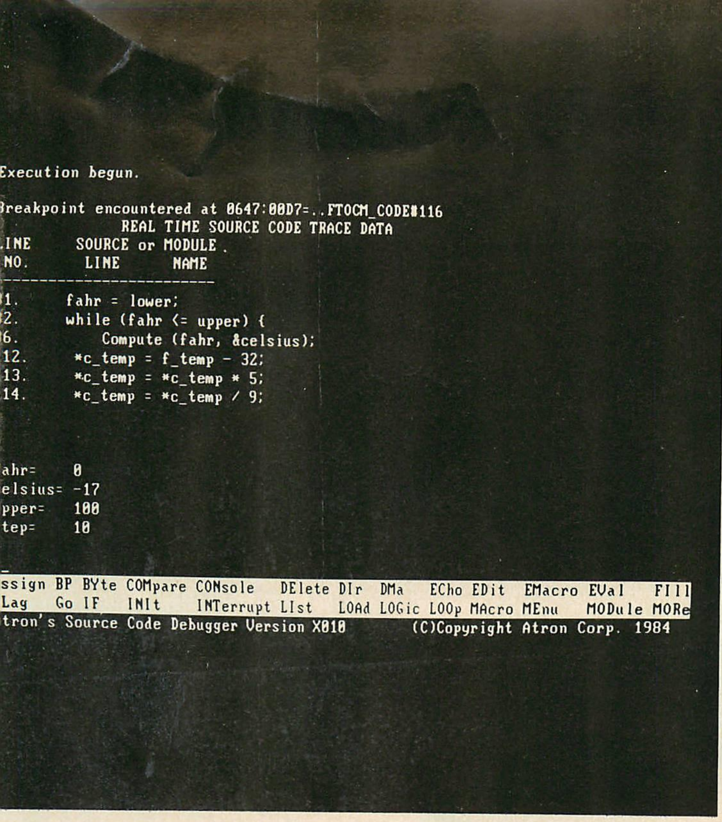
Even closer to PC-DOS is DOS Commander, which does not help with syntax. DOS Commander may make things go faster (even that is questionable), but it certainly does not make using DOS any easier.

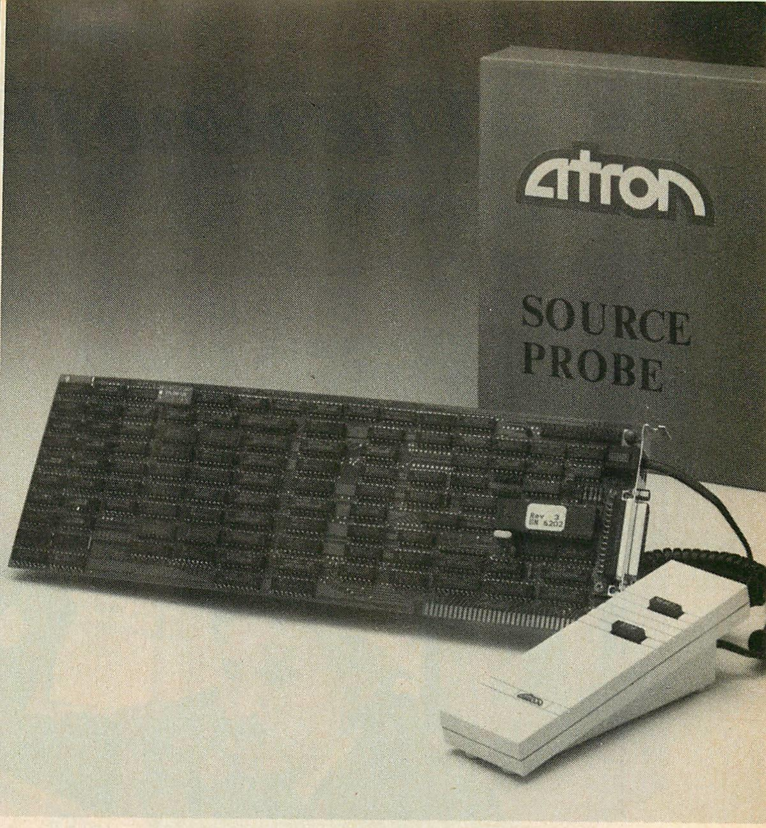
Finally, DOS Shell is not a shell at all; it is simply a collection of batch files and one BASIC program. Although DOS Shell does not at all simplify the use of PC-DOS, XT owners might be interested in this program for its ability to organize the fixed disk; they should, however, ignore the applications programs that come with it.

DOS Commander
Applitek
c/o Connecticut Software
30 Wilson Avenue
Rowayton, CT 06853
203-838-1844
$79.95
CIRCLE 500 ON READER SERVICE CARD

DOS Ease
Software Solutions, Inc.
305 Bic Drive
Milford, CT 06460
203-877-9268
$66.00
CIRCLE 499 ON READER SERVICE CARD

DOS Shell
XTC Handcrafted Software
P.O. Box 902
Santa Barbara, CA 93102
805-967-7837
$70.00
CIRCLE 498 ON READER SERVICE CARD





ATRON Announces A State of the Art Advance in Source Level Software Debugging

PROGRAMMERS AND MANAGERS know that finding bugs during new product development and over the entire product life cycle adds up to a significant portion of total product development cost and support time. Investing in the right debugging tools will greatly improve time to market as well as minimize development cost. Atron Corp. has the right debugging tools for the PC environment. These are:

1) PC PROBE
2) SOURCE PROBE
3) PERFORMANCE ANALYZER

These tools run on a PC or compatible.

A Snapshot of Real Time Program Execution
Program flow is saved in trace memory while running at full speed. PC PROBE can display trace data as high level language source code or as 8088 instructions. Real time trace answers the question "How did I get here"?

A Bullet of Proof Debugger
What good is a debugger that can be wiped out by an undebugged program? PROBE software is write protected and can't be changed.

Trap Memory Overwrites
The PC PROBE has 8 breakpoints and can trap conditions such as instruction execution, read, write, IO, DMA, interrupt, or external logic probes. Breakpoints can also be set on ranges of address or data — symbolically too!

Don't Look in the Manual
The PC PROBE designers know the importance of EASE OF USE. The PC PROBE interface has a menu window which displays the syntax of each command — so you never have to remember how a command works. It also recalls the previous invocation of each command to save tedious typing — and tedious thinking!

Symbolic Debugging
Avoid the tedium of sifting through link maps to find out where things are. The PC PROBE uses your C, PASCAL, or assembly language program symbols.

Macro Commands
Why be limited by a fixed set of debugging commands? PC PROBE lets you create your own powerful macro commands with parameter passing, nesting, LOOPING and IF/THEN/ELSE control.

ANNOUNCING SOURCE PROBE

Why waste time with a debugger which shows your C or PASCAL programs in 8088 assembly language. Atron's new SOURCE PROBE let's you step, real time trace, and edit your Hi Level source code during debug. Display data structures in the format which your program uses.

PERFORMANCE ANALYZER

How do you find time critical program problems or know where to start performance tuning your software? Get Atron's new Software Performance and Timing Analyzer. Then you can display historgrams of how your programs run — by time or by events. You can perform many different timing measurements.

Atron has many happy customers who have made critical product schedules because of PC PROBE. Why waste time on primitive debugging techniques? — Call us today and ask for your 12-page data sheet.

atron

20665 FOURTH STREET • SARATOGA, CA 95070 • (408) 741-5900

CIRCLE NO. 107 ON READER SERVICE CARD

The new IBM DisplayWrite

Little Tramp character licensed by Bubbles Inc., s.a.

*Prices apply at IBM Product Centers and may vary at other stores.

**Available in July, 1984.

Enhancements

SOFTWARE

Aptec *(for color Prism Printers)*
- Rainbow Writer Color Text Formatter .. $119.
- Rainbow Writer Screen Grabber ...... 69.

Ashton Tate
- dBase II ...... 279.
- dBase III ...... 375.
- Upgrade (dBase II to dBase III) ...... 139.
- Framework ...... 375.
- Friday! ...... 169.
- Reference Encyclopedia *(book)* ...... 55.

Best Programs
- Personal Finance Programs ...... 65.
- PC/Professional Finance Program II .. 149.
- PC/Fixed Asset System ...... 297.
- PC/Tax Cut ...... 149.

Bible Research
- THE WORD *(KJV Bible - 7 disks)* ...... 145.

Bruce & James
- WordVision ...... 39.

Digital Research
- CP/M-86 ...... 39.
- Dr. LOGO *(requires 192k & RGB monitor)* 69.

Financier
- Financier II ...... 115.
- Financier Tax Series ...... 97.

Funk Software
- Sideways ...... 45.

Lifetree
- Volkswriter ...... 115.
- Volkswriter International ...... 135.
- Volkswriter Deluxe *(with TextMerge)* ... 165.

Lotus Development
- 1-2-3 *(version 1A)* ...... call
- Symphony ...... call

Micropro
WordStar & Propak come with quick lesson CAI training disk and tutorial disk
- WordStar 3.3 ...... 219.
- Propak *(WordStar/MailMerge/CorrectStar/StarIndex)* ...... 279.

Microrim
- R:base 4000 ...... 279.
- Extended Report Writer ...... 85.
- Clout ...... 119.

Microsoft
- Multiplan *(ver 1.2)* ...... 139.
- Financial Statement *(for Multiplan)* ...... 69.
- Budget Analysis *(for Multiplan)* ...... 99.
- Microsoft Word *(ver 1.1)* ...... 239.
- Microsoft Word with mouse *(ver 1.1)* .. 289.

Microstuf
- Crosstalk XVI ...... 99.
- Infoscope ...... 149.
- Transporter *(includes Crosstalk)* ...... 169.

PCsoftware
- PCcrayon II *(new release)* ...... 39.
- Executive Picture Show ...... 139.
- CREATABASE ...... 47.

Peachtree
- PeachPak Series 4 (G/L, A/R, A/P) *new version, IBM manuals* ...... 195.

Peter Norton
- Norton Utilities ...... 55.

Rosesoft
- Prokey 3.0 ...... 89.

Satellite Software
- WordPerfect ...... call

PC Connection Software Special
through October 31, 1984

Software Publishing
PFS/Access
The first of a series of PFS telecommunications products.
Works with the following modems:
- Hayes Smartmodem 300, 1200,1200B
- Novation 103 and 103/212 Smart-Cat
- Transend PC Modem Card
- US Robotics Password
- Ven-Tel PC Modem Plus & others
- Included with the package are offers valued at over $200 from The Source, Dow Jones, Western Union Easylink, MCI Mail, Compuserve & Micro Communications Magazine ...... $59.

Software Publishing
- PFS/Access ...... see special
- PFS/File ...... $89.
- PFS/Graph ...... 89.
- PFS/Write ...... 89.
- PFS/Report ...... 79.

Softword Systems
- MultiMate *(ver 3.22 w/spell checker & tutorial)* ...... call

Sorcim
- Supercalc 3 ...... 199.

Virtual Combinatics
- Micro Cookbook ...... 29.

Warner Software
- The Desk Organizer *w/1 yr. free updates* ...... 197.

TRAINING

ATI
- How to use Lotus 1-2-3 *(volume 1 & 2)*. 49.
- How to use Wordstar *(volume 1 & 2)* ... 49.
- How to use dBase II *(volume 1 & 2)* ... 49.
- How to use EasyWriter II ...... 49.
- How to use Multiplan ...... 49.
- How to use Your IBM-PC ...... 32.
- How to use MultiMate *(volume 1 & 2)* .. 49.
- How to use Microsoft Word ...... 49.
- How to use PC DOS ...... 32.
- How to use TK!Solver ...... 49.

Individual Software
- The Instructor ...... 35.
- Professor DOS ...... 47.
- Tutorial Set *(both items above)* ...... 75.
- Professor Pixel ...... 47.
- Typing Instructor ...... 39.

Scarborough Systems (was Lightning)
- Master Type ...... 39.

EDUCATIONAL

Davidson
- Speed Reader II *(high school & college)* 49.
- Math Blaster *(grades 1-6)* ...... 35.
- Word Attack *(grades 4-12)* ...... 35.

Friendly Soft
- FriendlyWare/PC Introductory Set ...... 35.

Stone *(requires graphics board)*
Great graphics and sound
- My Letters, Numbers,Words *(ages 1 to 5)* 29.
- Kids Stuff *(agest 3 to 8)* ...... $29.
- Across the U.S.A. *(ages 5 & up)* ...... 22.

GAMES

Atarisoft
- Centipede ...... 29.
- Pac Man ...... 29.
- Dig Dug ...... 29.
- Defender ...... 29.
- Donkey Kong ...... 29.
- Stargate ...... 29.
- Robotron ...... 29.

Broderbund
- Apple Panic ...... 23.
- Lode Runner ...... 25.

Funtastic
- Snack Attack II *(a favorite)* ...... 27.
- Cossmic Crusader *(as good as above)* 27.
- Big Top *(climb to new levels)* ...... 29.
- Master Miner *(1 or 2 players)* ...... 29.

Hayden Software
- Sargon III ...... 35.

Microsoft
- Flight Simulator *(new version)* ...... 35.

Orion
- J-Bird *(you'll never sleep)* ...... 29.
- PC-Man ...... 25.
- Paratrooper ...... 25.
- Pits & Stones ...... 29.

PCsoftware
- Championship Blackjack ...... 23.

Sir-Tech
- Wizardry ...... 42.

Sublogic
- Night Mission Pinball ...... 29.

HARDWARE

AST Research (For IBM-PC or XT)
All AST Boards come with SuperDrive, SuperSpool, and one year warranty.
- SixPakPlus 64k upgradeable to 384k, with clock calendar, serial and parallel ports *(game port optional)* ...... 269.
- MegaPlus II 64k upgradeable to 256k (or more with MegaPak) with clock calendar and serial port *(parallel, game, or second serial port optional)* ...... 269.
- MegaPak 128k *(not upgradeable)* ...... call
- MegaPak 256k ...... call
- I/O Plus II with clock calendar and serial port *(parallel, game, or second serial port optional)* ...... 129.
- Parallel Game or second Serial Port *(for any AST board (specify board)* ...... 39.
- Connectall *connector bracket (PC only)* . 19.
- AST-5251 ...... 559.
- AST-3780 ...... 649.
- MonographPlus with clock calendar, serial & parallel ports ...... 429.

Amdek
- Video 300G monitor *(green)* ...... 149.
- Video 300A monitor *(amber)* ...... 159.
- Video 310A monitor *(amber)* ...... 179.

Compucable
- Plastic Keyboard & Drive Cover Set .... 17.
- IBM Mono Screen Enhancement ...... 17.
- Printer to IBM Cable *(specify printer)* ... 32.
- Smartmodem to IBM Cable ...... 25.

for advancements.

Curtis
STANDS
PC Pedestal *(for IBM Mono or Color)* . . . $39.
PGS or Quadchrome Adapter for above . . 9.
Low Profile Tilt and Swivel Pedestal . . . . 39.
System Stand . . . . 21.

CABLES
Extension Cables for IBM Mono Display 39.
Keyboard Extension Cable *(3 to 9 feet)* 29.
AC Plug Adapter *(any monitor to your PC)* 8.

SURGE SUPPRESSORS - NEW
. . . . See special

Epson
RX-80 with GRAFTRAX-Plus . . . . call
RX-80 F/T with GRAFTRAX-Plus . . . . call
FX-80 with GRAFTRAX-Plus . . . . call
RX-100 with GRAFTRAX-Plus . . . . call
FX-100 with GRAFTRAX-Plus . . . . call
LQ-1500 *(letter quality dot matrix)* . . . . . . call
Printer to IBM Cable (specify printer) . . . 32.

Hayes
Smartmodem 300 . . . . 209.
Smartmodem 1200 . . . . 489.
Smartmodem 1200B *(w/Smartcom II)* 409.
Smartcom II . . . . 99.
Compucable's Smartmodem-to-IBM Cable . . . . 25.

Hercules Computer
Hercules Graphics Card *(parallel port)* 339.
Hercules Color Card *(parallel port)* . . . . 179.
Graph-X Software . . . . 42.

key tronic
Typewriter style keyboard (KB 5150) . . . call
Deluxe keyboard (KB 5151) . . . . call

Koala
Koala Touch Tablet *with software (connects to game port)* . . . . 89.

Kraft
Joystick . . . . 39.

Maynard Electronics
Floppy Drive Controller . . . . 135.
Internal Hard Disk (10 Meg),WS-1 Controller & ROM for PC mother board . . . . 879.
Internal Hard Disk (10 Meg) WS-2 Controller & ROM for PC Mother Board . . . . 1079.

Mouse Systems
PC Mouse (w/software & desk pad) . . . 189.

NEC
Spinwriter 3550 *(IBM-PC compatible)* 1625.
Spinwriter 2050 *(3550's little brother)* . . 869.

Orchid Technologies
All Orchid Boards come with PCnet Drive (Ram disk). PCnet spool (print spooling), disk caching & partitioning.

Blossom 64k upgradeable to 384k, with clock calendar, serial and parallel ports 259.

PCnet Daughter Board (piggybacks to Blossom) with version 2.4 networking software . . . . call

Paradise Systems
MultiDisplay Card *(color & mono)* . . . . . 289.

Plantronics/Frederick
COLORPLUS *(w/Color Magic)* . . . . call

PC Connection Hardware Special
through October 31, 1984

CURTIS
Surge Protectors
Curtis Manufacturing has designed a new line of surge protectors to be competitive with the EPD Lemon, Lime, Orange and Peach. The Curtis protectors exceed EPD's specs and are covered by a lifetime warranty against defects.

- Diamond *(similar to EPD Lemon)* $35.
- Emerald *(similar to EPD Lime)* . . $45.
- Sapphire *(similar to EPD Peach)* $55.
- Ruby *(similar to EPD Orange)* . . . $65.

WHEN YOU CALL PC CONNECTION

In order to process your order quickly and efficiently, we need to feed our new mainframe computer a few tidbits of information. Bill to. Ship to. Method of payment. Method of shipment. And, of course, what you'd like to order.

We'd also like to know where you saw our ad and, if possible, we'd like you to read us the small code number that appears next to our phone number.

If you have all of this information ready when you call, we'll have you back to work and your products on the way that much quicker.

OUR POLICY

- We accept VISA and MASTERCARD.
- No surcharge added for charge cards
- Your card is not charged until we ship.
- No sales tax.
- All shipments insured; no additional charge.
- Allow 1 wk. for personal & co. checks to clear.
- COD max. $500. Cash or certified check.
- 120 day guarantee on all products.★
- To order, call us anytime Monday thru Friday 9:00 to 9:00, or Saturday 9:00 to 5:30.

SHIPPING

Continental US: For monitors, printers, and drives, add 2% to all orders. For all other items, add $2 per order for UPS surface, $3 per order for UPS 2nd-Day-Air. We particularly recommend 2nd-Day-Air if you live west of the Mississippi or south of Virginia. In most cases it will only cost $1 more and will save you up to four days on delivery time. **Outside Continental US:** We add freight charges to credit card purchases. For prepayment, call 603/446-3383 for information.

1-800/243-8088

PC Connection 295T
6 Mill St., Marlow, NH 03456
603/446-3383

Princeton Graphics
HX-12 RGB monitor *(690 x 240)* . . . . call
SR-12 RGB monitor *(690 x 480)* . . . . call
Scan Doubler Board (for SR-12) . . . . call
MAX-12 Amber monochrome monitor . . call

Quadram
We are a full line Quadram Dealer
New Expanded Quadboard 64k expandable to 384k, with clock calendar, parallel, serial & game port, I/O bracket, and Quadmaster software . . . . $269.
Microfazer Printer Buffer (parallel) w/copy
MP 64 (64k) upgradeable to 512k . . . . 197.
Quadcolor I . . . . 197.
Quadchrome RGB Monitor . . . . 489.
Quadlink *(allows you to run most Apple II programs directly on your IBM-PC or XT)* 459.

SMA (Systems Management)
PC-Documate Keyboard Templates
available for:

DOS/Basic 1.1	Multimate
DOS/Basic 2.0	Multiplan (IBM)
Lotus 1-2-3	Multiplan (Microsoft)
dBase II	Peachtext 5000
Easywriter II	Volkswriter

Wordstar . . . . *(each)* 12.

TG
Joystick . . . . 45.

USI *(monitors for graphic board)*
Pi-2 Monitor (12" green, with cable) . . . 119.
Pi-3 Monitor (12" green, with cable) . . . 129.

DRIVES

All drives are completely pre-tested. Specify Drive A or Drive B for your PC. Comes with complete step by step installation instructions. Drives are 320k/360k.

Tandon
TM 100-2 (5¼")*full-height drive (DS,DD)* 199.

TEAC
FD-55B (5¼") *half-height drive (DS,DD)* 169.
Free 'y' cable & bracket with each pair.

MEMORY

64k Memory Upgrade Set for IBM-PC or XT system board . . . . 59.
64k Memory Upgrade Set for any memory board *specify make of board* . . . . 59.
Install memory upgrades & run diagnostics *at time of board purchase only* . . . . 10.

DISKS

Verbatim *(with 5 year guarantee)*
Datalife Disks SS/DD *plastic box of 10* 23.
Datalife Disks DS/DD *plastic box of 10* 33.
Disk Drive Analyzer . . . . 22.
Flip Sort *(holds 75 disks)* . . . . 19.

INFORMATION SERVICES

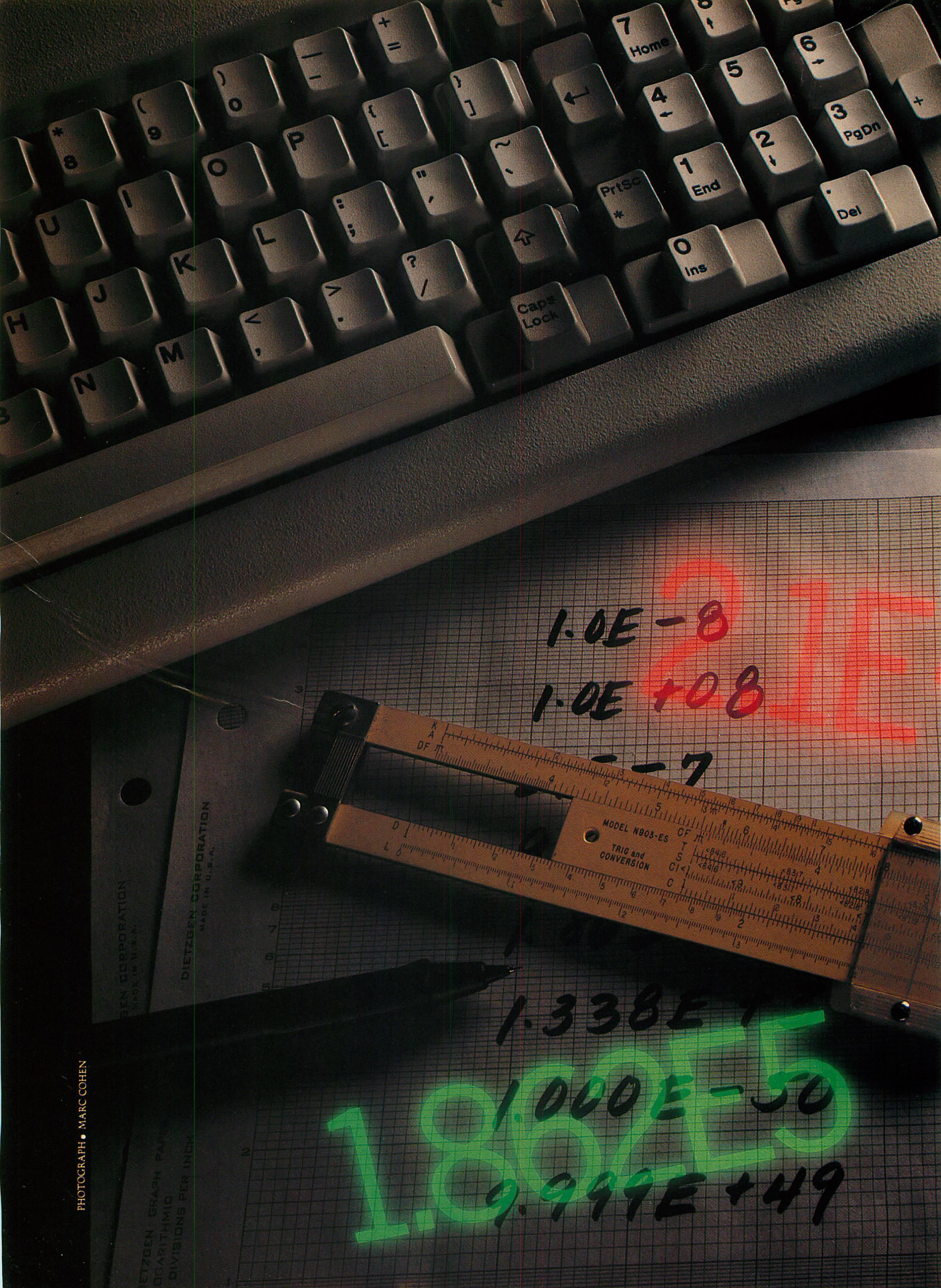
Compuserve
Compuserve Information Service *(includes subscription, manual, 5 hours of connect time, monthly publications)* . . . . 29.

Source Telecomputing
The Source *(subscription & manual)* . . . 49.

For the IBM-PC *Exclusively.*

PC CONNECTION®

CIRCLE NO. 182 ON READER SERVICE CARD



PHOTOGRAPH • MARC COHEN

Significant Figures, I

Floating-point addition and subtraction on the IBM PC

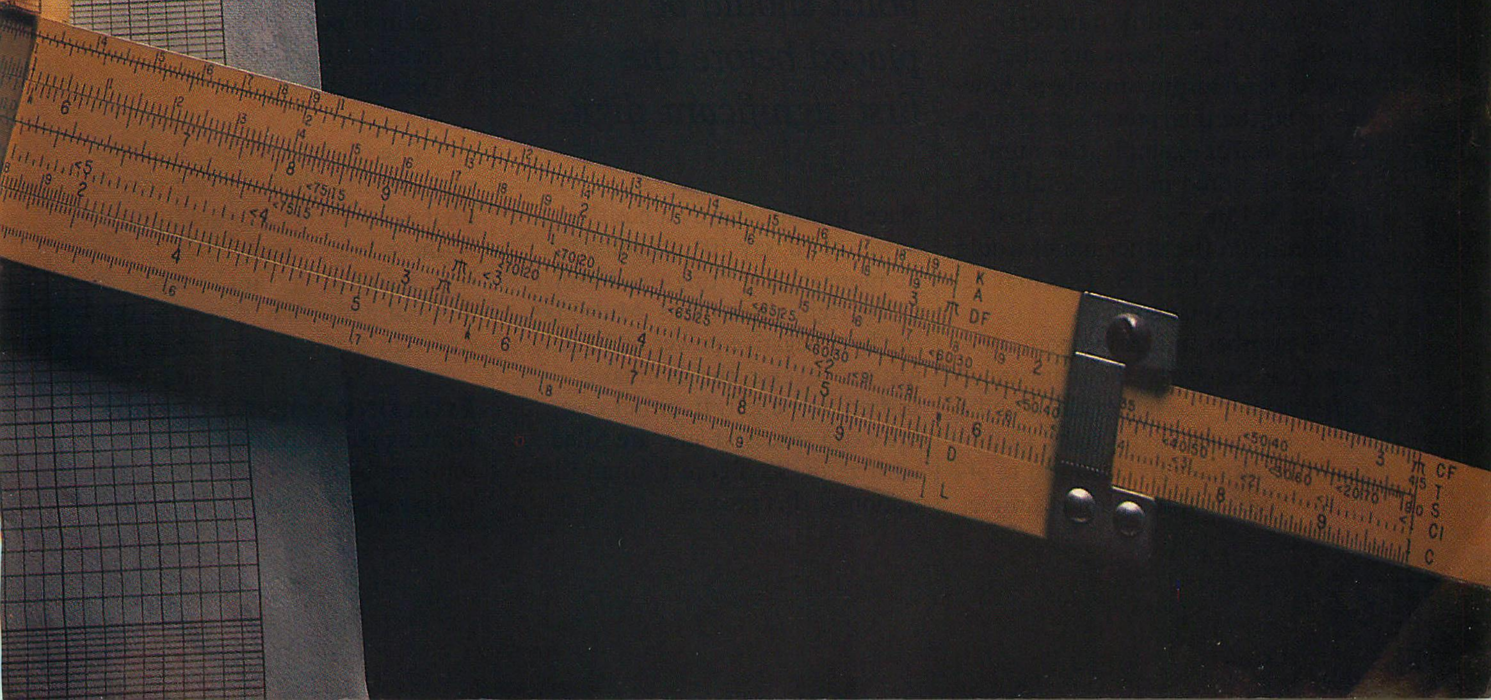
ROBERT GRAY

Because the Intel 8088 (the processor that does the computational work for the IBM PC) is a 16-bit integer processor, it cannot represent fractions or work with numbers larger than 65,535, the largest number representable in 16 bits. The ability to store and add fractions and very large numbers requires complex machine-language programs called floating-point routines. It is these routines that make arithmetic so simple in BASIC and virtually all other high-level languages.

As assembly language programmers know, the lack of floating-point routines enormously complicates the job of developing machine-language programs. Yet surprisingly few programmers have any notion of how floating-point routines work or more than the vaguest notion of what they do. Without special training, assembly

---

This is the first of three articles by Robert Gray on the subject of floating-point arithmetic on the IBM PC. Part two will discuss methods of using floating-point numbers to do multiplication and division; part three will cover binary-to-decimal floating-point conversion. Gray is an assistant professor of information systems at Virginia Commonwealth University.



language programmers who try to write their own routines for floating-point arithmetic face problems.

Reading the many published descriptions of floating-point representation will probably not help, because floating-point routines are very difficult to code and notoriously hard to debug. Understanding the procedures required for doing floating-point arithmetic is essential; published algorithms are as obscure as they are rare, however.

This article is intended to provide assembly language programmers with all that they need to know to do floating-point arithmetic on the IBM PC and any other microcomputer that uses the 8088 or 8086 microprocessor. It describes floating-point numbers and representation and gives an algorithm for doing the most difficult floating-point operations—which are, surprisingly, simple addition and subtraction. For those assembly language programmers who cannot or will not take the time to write their own routine, the article includes a tested and thoroughly documented assembly language routine for addition and subtraction.

FLOATING-POINT NOTATION

BASIC prints numbers in essentially two ways. For example, the number one million would be printed as 1000000; one millionth, as .000001. Such numbers are clear; they are what most of us think numbers should look like. There are other ways of representing numbers, however, and the interpreter sometimes uses them. For example, the number one hundred million would be printed as 1.0E+08; one hundred millionth, on the other hand, would be printed as 1.0E−08. This may require some explanation.

A number in floating-point notation has two parts: the *mantissa*, or *fraction*, to the left of the *E* and a signed *exponent* to the right. Converting a number in this form to ordinary positional notation is simple. Just multiply the mantissa by 10 raised to the power given as the exponent. For example, 1.86282E5 is equivalent to 1.86282×10^5, or 186,282—the approximate speed of light in miles per second. The number 2.1E−7 would be equivalent to 2.1×10^{-7}. Because 10_{-7} is equal to $1/10^7$ or to 0.0000001, the number is 0.000000210—the approximate length in seconds of a single clock cycle on the IBM PC.

Actually there is a simpler way. Multiplying any decimal number by 10 is equivalent to shifting the number left one place or to shifting the decimal point to the right one place; multiplying by a positive power of 10, say 10^3, amounts to no more than shifting the decimal to the right three times. Multiplying by a negative power of 10 is just the reverse; the decimal is shifted to the left. In either case, the extra spaces, if any, are filled with zeros. See figure 1 for an illustration of this process.

The convention that the decimal should follow the first non-zero digit is arbitrary. It could just as easily have been decided that, in a normalized floating-point number, the decimal point should be placed before the first significant digit.

Converting from floating-point to fixed-point notation is not difficult, but what about converting in the other direction? How, for example, would the decimal number 10 be represented as a floating-point number? It could be written equally well as .10E+2 or as 100.0E−1. Theoretically, at least, there is an infinite number of ways in which 10 could be expressed in floating-point notation, depending on where the decimal is placed.

For consistency, then, floating-point numbers are expressed in a conventional form. The decimal point follows the first significant digit. Thus, the decimal number 10 would normally be written as 1.0E+1, and numbers that are expressed according to this rule are said to be *normalized.*

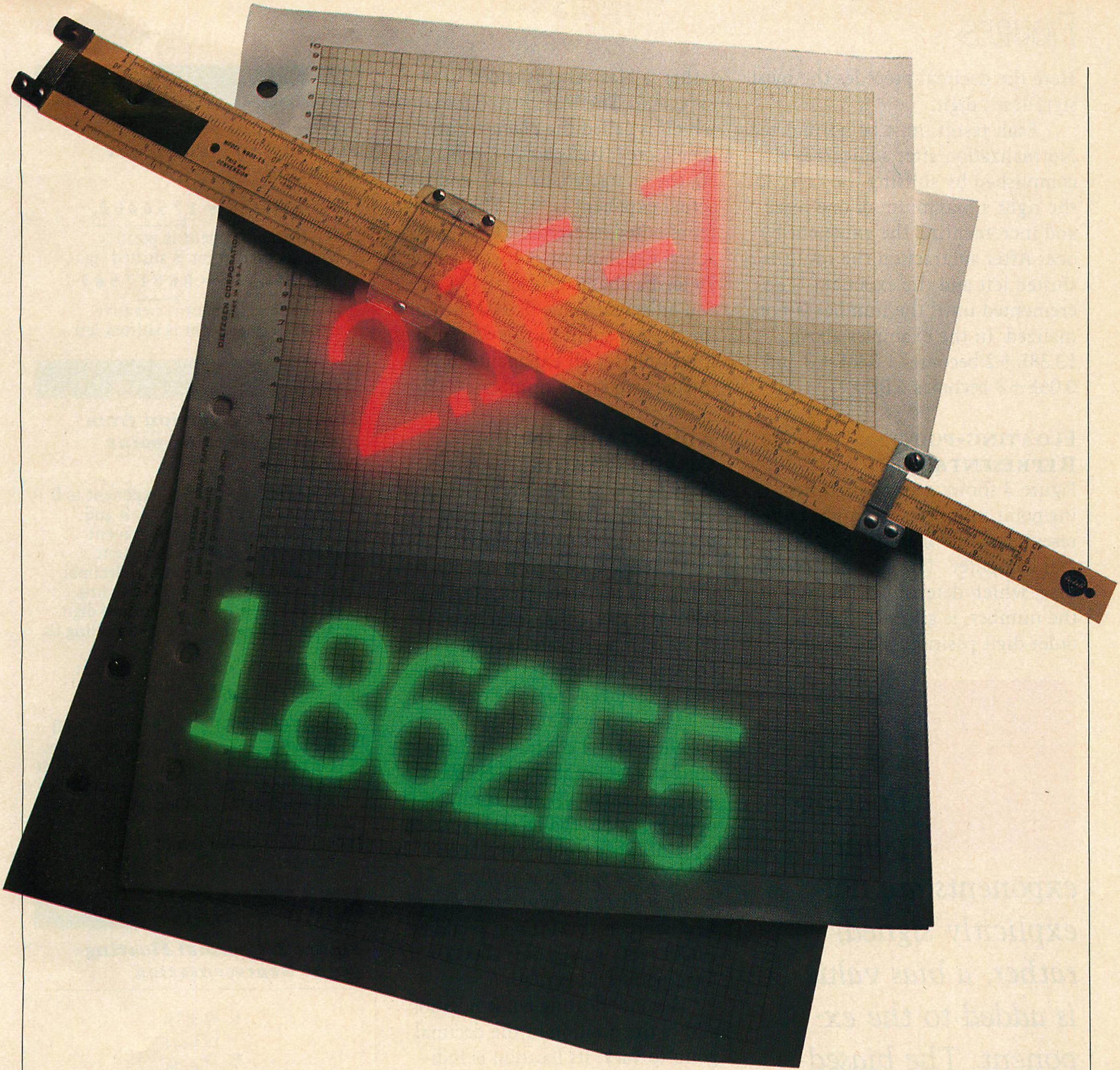
Once a convention governing where the decimal point should go has been established, it is possible to convert from fixed- to floating-point notation with the following steps (see also figure 2):

1. Set the exponent to zero.
2. If the decimal point is to the right of the first non-zero digit, shift the decimal to the left, incrementing the exponent, until the decimal point follows the first significant digit.
3. If the decimal point is to the left of the first non-zero digit, shift the decimal to the right, decrementing the exponent, until the decimal point follows the first significant digit.
4. Discard any leading or trailing zeros.

The convention that the decimal should follow the first non-zero digit is arbitrary. It could just as easily have been decided that, in a normalized floating-point number, the decimal point should precede the first significant digit.

FLOATING-POINT ADDITION

Floating-point notation presents some problems for arithmetic. It is not possible to add the following:



$$\begin{array}{lr} 9.9\text{E}-3 = & 0.0099 \\ +9.7\text{E}+2 = & 970.0 \end{array}$$

because the decimal points are in different places. To add any two floating-point numbers, the two decimal points must first be aligned.

This can be done by denormalizing the number with the smaller exponent, shifting it to the right, and incrementing its exponent a sufficient number of times to make the exponents equal. Note that shifting the mantissa right is equivalent to dividing it by 10, so that incrementing the exponent does not change the expression's value. In the example above, the smaller exponent is -3, the larger is 2, and the difference is 5. Moving the decimal point five places to the left in the smaller number gives

$$\begin{array}{ll} 0.000099\text{E}+2 = & 0.0099 \\ +9.7\text{E}+2 & = 970.0 \\ \hline 9.700099\text{E}+2 & = 970.0099 \end{array}$$

Normalization

Ordinarily, the exponent of the result is the exponent of the original larger number, but because floating-point numbers are normalized, this will not always be the case. It is possible for the addition of two floating-point numbers to yield an unnormalized result. For example,

$$\begin{array}{r} 5.79\text{E}+2 \\ +\ \ 7.59\text{E}+2 \\ \hline = 13.38\text{E}+2 \end{array}$$

The result is not considered normalized, because the decimal point does not follow the most significant digit.

Subtraction can also yield an unnormalized result. For example:

$$\begin{array}{r} 1.50\text{E}+3 \\ -1.44\text{E}+3 \\ \hline =0.06\text{E}+3 \end{array}$$

Here the decimal precedes the most significant digit.

Each result must be normalized. Normalization after addition is accomplished by shifting the carry to the right a maximum of one place and incrementing the exponent by one. After subtraction the result is shifted left and the exponent is decremented until the number is normalized. In the examples above, 13.38E+2 becomes 1.338E+3 and 0.06E+3 becomes 6.0E+1.

FLOATING-POINT REPRESENTATION

Figure 3 shows a seven-digit floating-point number as it might be represented in the memory of a decimal computer. The sign of the mantissa, which determines the sign of the number, is given in the high-order digit position. The next two digits represent the exponent. The lower four digits are reserved for the mantissa. In normalization the decimal point need not be stored; it may be assumed that it follows the first significant digit.

The exponent's sign is not shown. As a rule, floating-point exponents are not explicitly signed; rather, a bias value is added to the exponent. The biased exponent, which is called the *characteristic*, is then stored. In figure 3, the bias is 50, the characteristic is 52, and the actual exponent is +2. The number represented, therefore, is 317.8. An actual exponent of −2 would have yielded a characteristic of 48 (50−2) and an actual value of .03178. Storing the exponent in this way makes it possible for all exponents to be treated as positive numbers.

As a rule, floating-point exponents are not explicitly signed; rather, a bias value is added to the exponent. The biased exponent, which is called the characteristic, *is then stored.*

ADVANTAGES AND DISADVANTAGES

Why is floating-point representation used? After all, it seems to be a complicated way to store numbers.

There are two good reasons for using this notation. First, floating-point representation increases the range of representable numbers. Look again at figure 3. When the bias is removed from the exponent, the largest possible number is 9.999E+49. By contrast, had the same storage locations been used to store a number in ordinary fixed-point notation, the largest representable number would have been only 999999, not quite one million. The smallest floating-point number representable in this format would be 1.000E−50, versus only .000001 in fixed-point representation.

Second, floating-point notation allows the position of the decimal point to vary. Its location is indicated by the exponent. Suppose we used a fixed-point scheme that stored the whole part of a number in the upper three digits and the fraction in the lower three. The largest representable number would be 999.999, but we could not store 99.9999, although it is smaller and contains the same number of digits.

Floating-point representation has its disadvantages, however. Although this method does increase the range of representable numbers, it places certain constraints on their accuracy. Most single-precision floating-point routines provide six or seven digits of accuracy, meaning that only the first six or seven digits of any number are actually stored (the format shown in figure 3 stores only four digits). A number such as 1.245E+8 could thus be in error by as much as 49,999. So far as the floating-point routines are concerned, the numbers 124,549,999 and 124,450,001 are equal.

Figure 1: Conversion from Floating- to Fixed-point Notation

1.7564E6 = 1 7 5 6 4 0 0.

If the exponent is positive, the decimal point is shifted right

1.7564E−3 = 0.0 0 1 7 5 6 4

If the exponent is negative, the decimal point is shifted left

Figure 2: Conversion from Fixed- to Floating-point Notation

1756400.	E0	Set exponent to 0
175640.0	E1	Shift left and
17564.00	E2	increment
1756.400	E3	exponent
175.6400	E4	until decimal
17.56400	E5	follows first
1.756400	E6	non-zero digit
1.7564E6		Discard trailing 0s

If decimal is to the right of the first non-zero digit, shift it left

.0017564	E0	Set exponent to 0
0.017564	E−1	Shift right and dec-
00.17564	E−2	rement exponent
001.7564	E−3	until normalized
1.7564E-3		Discard leading 0

If decimal is to the left of the first non-zero digit, shift it right

Figure 3: Decimal Floating-point Representation

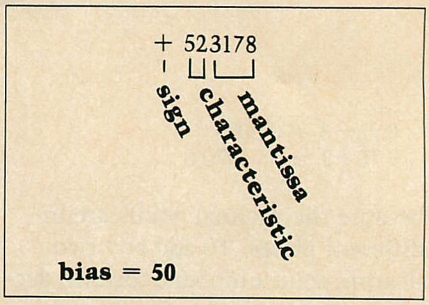


Figure 4: Binary Floating-point Formats (for +13.5)

sign 1 bit	exponent 8 bits	mantissa 23 bits
0	10000010	1011000 00000000 00000000

exponent bias=127

8087 floating-point format

exponent 8 bits	sign 1 bit	mantissa 23 bits
10000100	0	1011000 00000000 00000000

exponent bias=128

IBM PC BASIC (8088) floating-point format

Figure 5: Alignment and Rounding

operand1	10110001	00000011	00101000			exponent=134
operand2	10001011	10000010	00001101			exponent=131

operand1	10110001	00000011	00101000	
operand2	00010001	01110000	01000001	101

guard bit

Alignment without sticky bit
After alignment, the mantissa may be up to 48 bits long.

operand1	10110001	00000011	00101000		
operand2	00010001	01110000	01000001	1	1

guard bit sticky bit

Alignment with sticky bit
Only the guard bit is kept. A sticky bit indicates whether any ones were lost.

sum	11000010	01110011	01101001	1	1	
round +					1	
result	11000010	01110011	01101010			exponent=134

Round sum by adding one to guard bit.

This kind of inaccuracy is just the sort of problem that makes accountants have nightmares. Obviously, it is important that programmers take these limitations into account when they are developing procedures that utilize floating-point arithmetic, especially if it is possible that the error might be cumulative.

Binary Floating-point Representation

Several different floating-point formats are used on contemporary computers. Figure 4 shows two commonly used single-precision formats. The top one is the format used in the 8087 floating-point coprocessor. The other is the format used by IBM PC BASIC. The same number (+13.5) is shown in each case. Each format uses 32 bits. Although only 23 bits are shown, each format actually uses a 24-bit mantissa.

Floating-point numbers are stored in a normalized form so that the most significant digit is not zero. In binary, this necessarily means that the leading digit is a 1. Since the value of this digit is known, there is no need to waste space by storing it. The space saved is used to store the sign.

The two formats shown in figure 4 differ in some respects. On the 8087, the binary point is assumed to *follow* most significant digit, whereas on the 8088 the binary point is assumed to *precede* the leading digit. Thus, in the 8087 format, the mantissa is always greater than or equal to 1 and less than 2. In the BASIC format, the mantissa is always less than 1 and greater than or equal to .5 (0.1 in binary). If the leading ones are restored, the mantissas shown in figure 5 could be written as 1.1011 and 0.11011.

The two formats also use different exponent biases. In PC BASIC, the bias is 128, so the largest possible exponent is 127. Since the largest number representable in 8 bits is 255, a larger exponent would cause *overflow*, meaning that the number exceeds its allowable range. Similarly, the largest negative exponent is −128. A lesser exponent would produce a characteristic less than zero, causing *underflow*.

The 8087 uses a bias of 127. It would be logical to expect this to yield a maximum positive exponent of 128 and a maximum negative exponent of −127; however, the 8087 reserves the values 0 and 255, so the permissible exponent range is actually −126 to 127.

To convert the numbers shown in figure 4 to their decimal equivalents, subtract the exponent bias, move the binary point the appropriate number of places to the right of its implied position, and convert the resulting binary number to deci-

mal. Bear in mind that places to the right of the binary point express negative powers of 2; that is, 0.1 is equivalent to .5 decimal and 0.01 is equivalent to .25 decimal, and the binary fraction 0.01011 would be 11/32 (1/4+1/16+1/32).

Rounding

As noted above, floating-point addition requires that the operands be aligned by shifting the smaller operand to the right until the exponents are equal. This requirement creates a problem. Figure 5 shows two floating-point mantissas. The difference in the exponents is 3, meaning that the smaller operand is to be shifted right three bits. This will put 0s in the upper three bits, but what becomes of the lower three bits? It is tempting just to drop them, since they cannot add directly to the 24-bit mantissa of the result.

There are two schools of thought here. According to one school, all right-shifted digits should be kept to ensure the accuracy of intermediate results. This method would require that 48 bits be set aside for the mantissa of the smaller operand, because it might be shifted to the right as much as 24 bits. If alignment requires shifting more than 24 bits, the smaller operand would effectively become 0, since it could not affect the final sum. The other school of thought says it is sufficient to keep just enough information about the right-shifted bits to provide for accurate rounding.

Figure 5 illustrates this rounding technique. Here the mantissa of the right-shifted operand has been expanded to 26 bits. The leftmost extra bit is called a guard bit; it holds the last bit shifted out of the original mantissa. The other extra bit, called a sticky bit, serves a slightly different purpose. The sticky bit is on if any 1s have been shifted past the guard bit. When the mantissa is right-shifted, the guard bit is ORed into the sticky bit; the least-significant digit in the mantissa is shifted into the guard bit.

Addition is probably the most difficult of floating-point operations, and coding a floating-point addition routine on the 8088 is quite a challenge. The basic algorithm is, however, fairly straightforward.

```
1. Let E1, E2, S1, S2, M1, and M2 be the exponents, signs, and
   mantissas of the two operands.

2. Restore leading 1s

3. IF operand1 < operand2 THEN exchange E1,E2; M1,M2; S1,S2
               {ensure that larger number is in operand1
   ELSE
      IF operand1=operand2 and S1<>S2 THEN RESULT=0: GOTO EXIT
               {If the signs are opposite, operand2 is to be
               {subtracted from operand1.  This would yield
               {0 for equal operands (E1=E2 and M1=M2).

4. COUNT=E1-E2  {COUNT is the difference between the exponents

5. IF COUNT > 24 THEN RESULT=operand1: GOTO EXIT
               {Shifting the smaller operand more than 24 bits
               {to the right makes its mantissa equal to 0
   ELSE
      shift M2 right COUNT bits
               {The operands are aligned by shifting the smaller
               {right a sufficient number of places to make
               {the exponents equal.  At this point, two guard
               {bits (the last two bits shifted) are needed. Other
               {right-shifted bits are ORed into the sticky
               {bit.

6. IF S1=S2 THEN
      M1=M1+M2
               {If signs are equal, add the two operands.
      IF M1 > .5 THEN shift M1 right 1 bit: E1=E1+1
               {Addition may produce a carry to the left of
               {the binary point, so NORMALIZE
   ELSE
      M1=M1-M2
               {Opposite signs, so subtract operand2
      WHILE M1 < .5  shift M1 left 1: E1=E1-1 WEND
               {NORMALIZE M1 by shifting left, decrementing
               {the exponent, until the leading bit is a one

7. IF guard bit=1 AND (sticky bit is on OR M1 is odd) THEN M1=M1+1
               {Round to even

8. IF M1 > 1 THEN shift M1 right 1 bit: E1=E1+1
               {Normalize again if upward rounding produced
               {a carry to the left of the binary point

9. Set RESULT to E1, S1, M1

10. EXIT
```

Figure 6: An Algorithm (in Pseudo-code) for Floating-point Addition

Ordinarily, rounding is accomplished by adding .5 to the leftmost digit to be discarded. This technique introduces an upward bias, since the number is rounded up when the remainder is greater than or equal to .5 and rounded down only when the remainder is less than .5. For example, to round 98.9554 to two places, .005 would be added, producing 98.96. The number 98.9550, however, would also be rounded up. An unbiased technique would round a remainder equal to .5 upward half of the time and downward half of the time.

That is where the sticky bit comes in. If the guard bit is 1 and the sticky bit is 1, the remainder is greater than .5; the number will therefore be rounded upward. If the guard bit is 0, the remainder is less than .5, and the number will be rounded downward. If the guard bit is 1 and the sticky bit is 0, the remainder is equal to .5; the number will be rounded to the nearest even number—upward if the mantissa is odd and downward if it is even.

In figure 5, the number is rounded upward by adding 1 to the guard bit. It would have been just as easy to round to the nearest odd number, but rounding to the even number gives nicer results.

Figure 5 illustrates the situation just prior to rounding the result of an addition or subtraction operation. A sticky bit and one guard bit provide sufficient information for accurate rounding. In addition, two guard bits are kept during the actual addition or subtraction process. The sticky bit is then ORed into the second guard bit just before rounding.

An Algorithm for Floating-point Addition

Addition is probably the most difficult of all floating-point operations, and coding a floating-point addition routine on the 8088 is a challenge. The basic algorithm is, however, fairly straightforward (see listing 1).

Note that the assembly language routine given in listing 1 uses the IBM PC BASIC format, and the operands are passed in the same way. The result is returned in AX:DX. An advantage of this routine is that it is about 20 percent shorter and, presumably, proportionately faster than the routine used in the IBM PC BASIC compiler.

The algorithm shown in figure 6 closely resembles the algorithm used in listing 1; however, it is intended to be essentially machine- and language-independent.

These algorithms, along with the preceding discussion, should provide programmers with all they need to know to do floating-point arithmetic on their IBM PCs.

Listing 1: FLOAT.ASM

```
FLOAT_A  PROC        NEAR
                           ;entry point for subtraction
SUB_F:   MOV         BX,0080H    ;set mask to reverse sign of operand2
         JMP SHORT F1            ;and go to main routine
;
                           ;entry point for addition
ADD_F:   SUB         BX,BX       ;clear mask register BX
;
                           ;get operand1 from address in DI
F1:      PUSH        SI          ;save index registers
         PUSH        DI          ;
         MOV         DX,[DI]+2   ;exponent goes in DH--mantissa in DL
         MOV         AX,[DI]     ;and AX

                           ;get operand2 from address in SI
         XOR         BX,[SI]+2   ;exponent goes in BH--mantissa in BL
         MOV         CX,[SI]     ;and CX

            ;****************************************************
            ;        Floating point addition routine begins here
            ;        operands are in the following format:
            ;                DH        DL bit 7        DL AX

        ; Operand1  exponent    sign        mantissa

            ;
            ;                BH        BL bit 7        BL CX
            ; Operand2  exponent    sign        mantissa
            ;****************************************************

                           ;save signs
        MOV         SI,10000000B  ;put mask for sign bit is SI
         MOV         DI,SI        ;and in DI

         AND         SI,DX        ;put sign of operand1 in SI bit 7
         AND         DI,BX        ;put sign of operand2 in DI bit 7

         CMP         SI,DI        ;
         JE          A1           ;if signs are unequal
         ADD         DI,8000H     ;set bit 15 to 1.  Bit 15 is
         ADD         SI,8000H     ;addition/subtraction flag

A1:      OR          DL,10000000B ;restore leading ones
         OR          BL,10000000B ;
                                  ;put smaller operand in DX:AX
         CMP         DX,BX        ;compare first word
         JA          A2           ;exchange if DX:AX is larger
         JB          A3           ;

         CMP         AX,CX        ;equal first words; compare second
         JA          A2           ;

         JB          A3           ;
                                  ;operands are equal
         OR          DI,DI        ;if signs are unequal
         JNS         A3           ;

         SUB         DX,DX        ;set result in DX:AX to zero
         MOV         AX,DX        ;
         JMP         EXIT         ;and return

A2:      XCHG        DX,BX        ;exchange operands
         XCHG        AX,CX        ;
         MOV         DI,SI        ;put sign of larger in DI

A3:      MOV         SI,CX        ;larger operand is now in BX:SI
                                  ;compute exponent difference
```

```
        MOV    CL,BH       ;put larger exponent in CL
        SUB    CL,DH       ;and subtract the smaller exponent

                           ;CL is COUNT of bits to shift DX:AX
        CMP    CL,24       ;if COUNT > 24, smaller operand is
        JLE    A4          ;effectively 0, so
        MOV    DX,BX       ;put larger operand(result) in DX:AX
        MOV    AX,SI       ;
        JMP    FINISH      ;and finish
                           ;ALIGN fractions (shift smaller right)
A4:     SUB    DH,DH       ;clear DH
        MOV    CH,DH       ;clear CH

A5:     CMP    CL,8        ;
        JL     A6          ;if COUNT >= 8
                           ;shift right in byte increments
        OR     CH,DH       ;put trailing bits in CH
        MOV    DH,AL       ;move leading bits in AL (and any
                           ;others that happen to be set) to DH
        MOV    AL,AH       ;shift upper bytes right
        MOV    AH,DL       ;
        SUB    DL,DL       ;set leading byte to zero
        SUB    CL,8        ;decrement COUNT
        JMP    SHORT A5    ;if COUNT >= 8, repeat

A6:     OR     CH,CH       ;if any bits set in CH
        JZ     A7          ;
        OR     CH,80H      ;set sticky bit

A7:     OR     CL,CL       ;
        JZ     A9          ;if COUNT > 0
                           ;shift right COUNT bits
A8:     SHR    DL,1        ;rotate mantissa in DL:AX right
        RCR    AX,1        ;
        RCR    DH,1        ;rotate round bits in DH:CH right
        RCR    CH,1        ;
        DEC    CL          ;decrement COUNT
        JNZ    A8          ;repeat if COUNT in CL not zero
        MOV    CL,3FH      ;put mask in lower 6 bits of CL
        AND    CL,DH       ;pick up rounding bits from DH
        AND    DH,0C0H     ;zero out all but two guard bits in DH
                           ;operands are now aligned
A9:     OR     DI,DI       ;if signs were the same,i.e., sign
        JS     A10         ;bit = 0, then

        ADD    AX,SI       ;ADD the two operands
        ADC    DL,BL       ;
        JNC    R0          ;if there was a carry, fraction is
                           ;greater than 0, so normalize
        INC    BH          ;increment exponent
        JZ     OVER_F      ;jump if exponent overflow
        RCR    DL,1        ;shift fraction right one bit
        RCR    AX,1        ;
        RCR    DH,1        ;
        JMP SHORT R0       ;
                           ;SUBTRACT the two operands
```

```
A10:    XCHG   SI,AX       ;put larger operand in DL:AX
        XCHG   DL,BL       ;
        NEG    DH          ;subtract DH (guard bits)
        SBB    AX,SI       ;subtract significant portions
        SBB    DL,BL       ;
        JS     R0          ;if DL bit 7 is 1, round
                           ;NORMALIZE
A11:    DEC    BH          ;decrement exponent
        JZ     UNDER_F     ;jump if exponent underflow
        SHL    DH,1        ;shift mantissa left
        RCL    AX,1        ;
        RCL    DL,1        ;
        OR     DL,DL       ;
        JNS    A11         ;repeat if still not normalized

R0:     OR     CH,CL       ;set sticky bit before rounding
        JZ     R1          ;if any trailing bits in CX
        OR     DH,01000000B ;set sticky bit in DH
                           ;ROUND normalized result
R1:     OR     DH,DH       ;if guard bit in DH is 0,
        JNS    FINISH      ;rounding is not necessary
                           ;guard bit is on
        TEST   AL,1        ;if fraction is odd, then
        JNZ    R2          ;round it

        AND    DH,7FH      ;check trailing bits in DH and CX
        JZ     FINISH      ;if all zeros, rounding not needed

R2:     ADD    AX,1        ;round by adding one
        ADC    DL,0        ;
        JNC    FINISH      ;rounding may denormalize mantissa

        RCR    DL,1        ;normalize by shifting right one
        RCR    AX,1        ;
        INC    BH          ;increment exponent
        JZ     OVER_F      ;jump if exponent overflow


FINISH: AND    DL,7FH      ;set sign bit to 0
        OR     DX,DI       ;restore sign from bit 7 of DI
        MOV    DH,BH       ;put exponent in DH
        SUB    BX,BX       ;set status=0 -- OK
EXIT:   POP    DI          ;restore index registers
        POP    SI          ;
        RET

OVER_F: MOV    DX,0FF7FH ;here for overflow--set result in DX:AX
        MOV    AX,0FFFFH ;to maximum possible value with positive
        OR     DX,DI     ;sign. Restore actual result sign.
        MOV    BX,1      ;set status=1 -- overflow
        JMP SHORT EXIT   ;
UNDER_F: SUB   DX,DX     ;set result in DX:AX to zero
        MOV    AX,DX     ;
        MOV    BX,-1     ;set status=-1 -- underflow
        JMP SHORT EXIT   ;
FLOAT_A ENDP
```

TECH NOTEBOOK 24

Talking to a Mouse from Turbo Pascal

Using Turbo's INTR routine to make mouse function calls

JEFF DUNTEMANN

When I got Turbo Pascal V2.0, which has some graphics capabilities, my first test was to make it talk to the Microsoft Mouse. It turned out to be fairly easy, thanks to a marvelous Turbo Pascal built-in routine called INTR. This routine allows the tripping of any software interrupt from within a Pascal program. Turbo Pascal predeclares INTR in the following way:

PROCEDURE INTR(INTERRUPT : <INT_CONST>; VAR REGISTERS : REGPACK);

INTR takes two parameters: the number of the software interrupt and a record that contains register and flag values to be passed to the interrupt routine to be tripped. <INT_CONST> must be an integer *constant*, not a variable. REGISTERS is a record type that must be explicitly declared:

TYPE REGPACK = RECORD AX,BX,CX,DX,BP,SI,DI,DS, ES, FLAGS : INTEGER END;

Any values to be passed to the interrupt routine must be assigned to the appropriate field in REGISTERS before the interrupt is tripped. If the interrupt routine returns any values in the registers, these values are available when the interrupt returns control to the program.

Making a call to the Microsoft Mouse driver routine is done by placing parameter values into the AX, BX, CX, and DX registers and tripping software interrupt 51 (33 hex). The number of the function calls goes into AX; the remaining three registers hold optional parameters that depend on the mouse function call desired. Return values from the function call (if there are any) come back in the same registers. The Microsoft Mouse User Guide describes the parameters and return values fairly well.

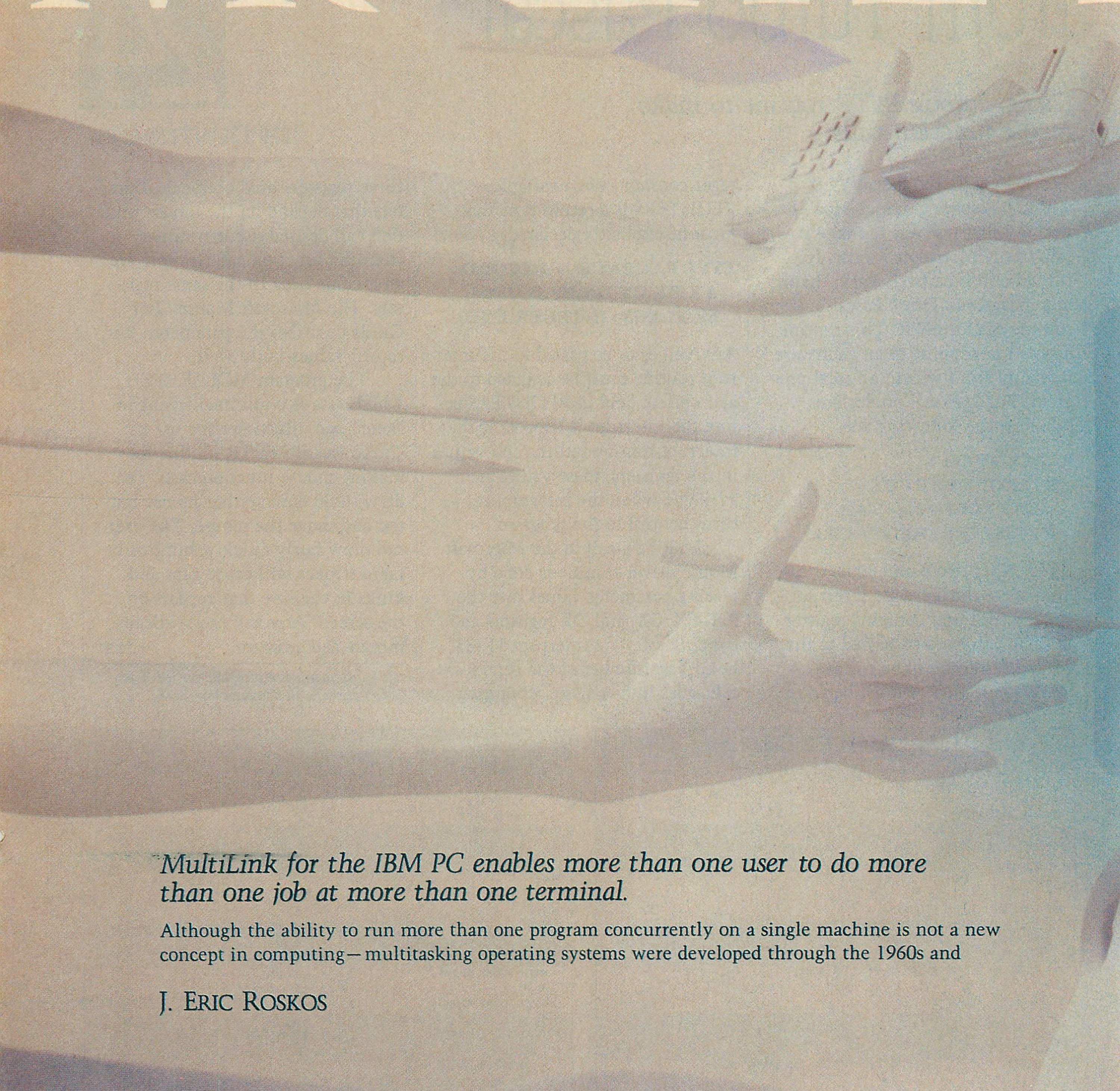
The program MOUSKETCH, which acts as a minimal freehand sketch pad, illustrates the use of Turbo Pascal's INTR routine for making mouse function calls. To draw, hold down either mouse button and move the mouse. The user can draw fairly quickly, but overly rapid strokes will cause gaps and kinks in the line that appears on the screen. Any key press will terminate the program.

Jeff Duntemann is a programmer for Xerox Corporation in Rochester, New York.

Listing 1: MOUSKETCH.PAS

```
{      MOUSKETCH       }
{   by Jeff Duntemann  }

PROGRAM MOUSKETCH;

TYPE REGPACK = RECORD
                 AX,BX,CX,DX,BP,SI,DI,DS,ES,FLAGS : INTEGER
               END;

VAR OLDX,OLDY,X,Y : INTEGER;
    M1,M2,M3,M4   : INTEGER;
    REGSET        : REGPACK;

PROCEDURE MOUSE(VAR M1,M2,M3,M4 : INTEGER);

VAR REGSET : REGPACK;

BEGIN
  WITH REGSET DO
    BEGIN            {Set up AX/BX/CX/DX for interrupt}
      AX := M1;
      BX := M2;
      CX := M3;
      DX := M4
    END;
  INTR(51,REGSET);   {Trip interrupt 51}
  WITH REGSET DO
    BEGIN            {Put return values back into M1-M4}
      M1 := AX;
      M2 := BX;
      M3 := CX;
      M4 := DX
    END
END;

BEGIN
  HIRES;                              {Choose graphics mode & color}
  HIRESCOLOR(YELLOW);
  M1:=0;                              {Init mouse driver}
  MOUSE(M1,M2,M3,M4);
  M1:=1;                              {Turn mouse cursor on}
  MOUSE(M1,M2,M3,M4);
  M1 := 3; OLDX := 0; OLDY := 0;
  WHILE NOT KEYPRESSED DO             {Exit MOUSKETCH when any key pressed}
    BEGIN
      MOUSE(M1,M2,M3,M4);
      IF M2 <> 0 THEN DRAW(OLDX,OLDY,M3,M4,1); {Draw if button pressed}
      OLDX := M3;                     {Update old X/Y values}
      OLDY := M4;
    END;
END.
```

MULTITA

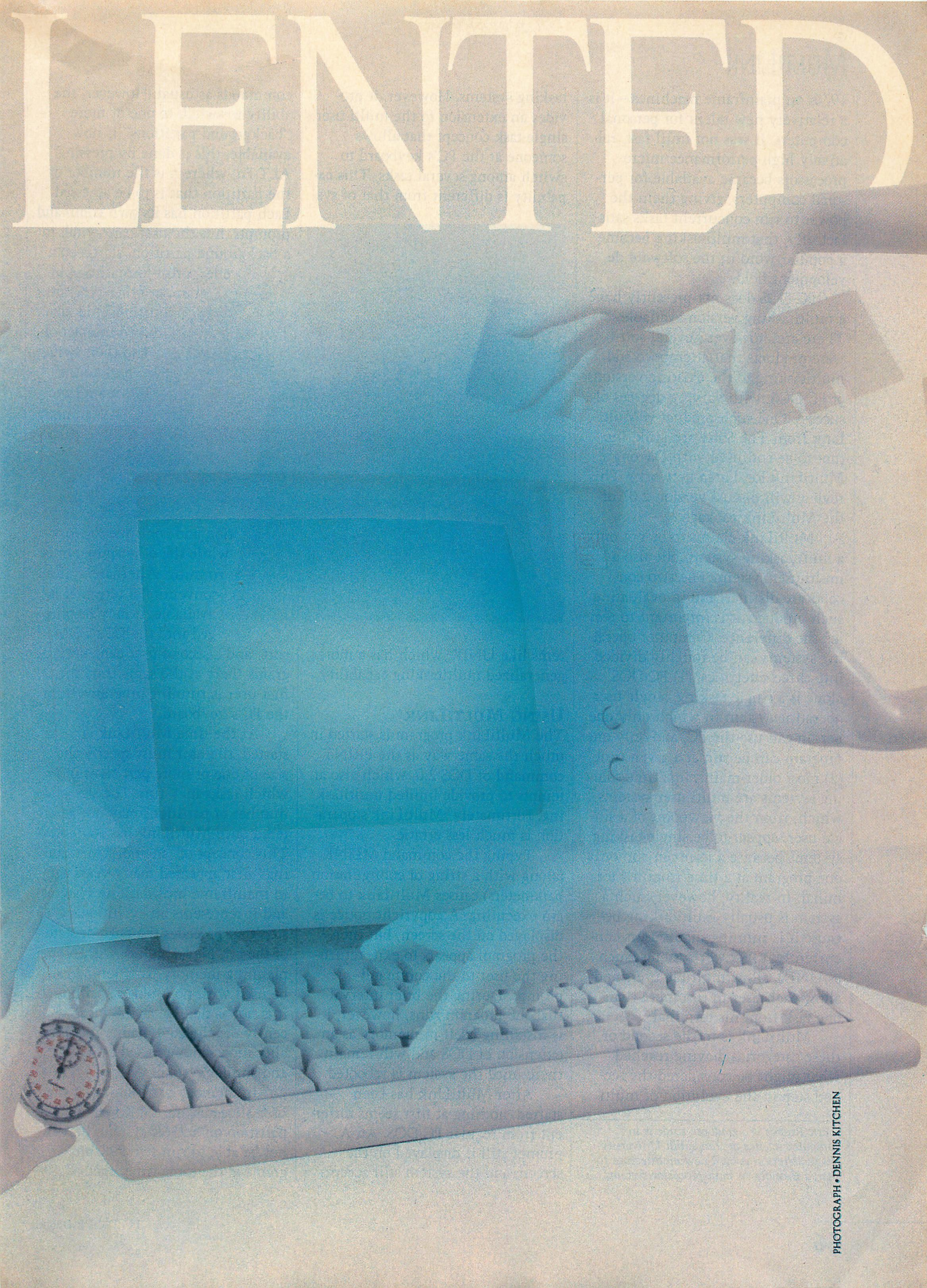


MultiLink for the IBM PC enables more than one user to do more than one job at more than one terminal.

Although the ability to run more than one program concurrently on a single machine is not a new concept in computing—multitasking operating systems were developed through the 1960s and

J. Eric Roskos

LENTED



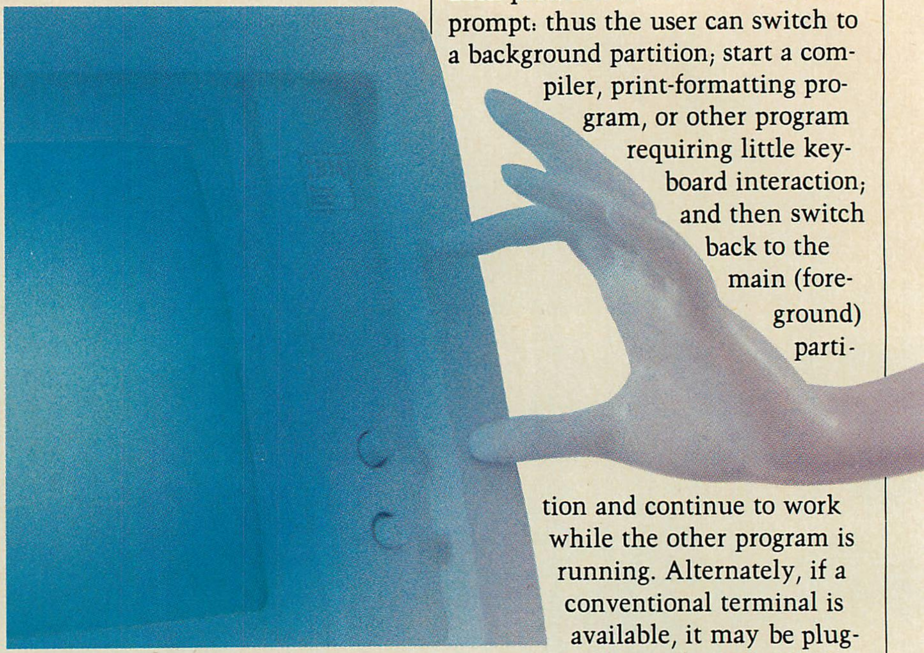
PHOTOGRAPH • DENNIS KITCHEN

1970s on mainframe machines—it is a relatively new talent for personal computers. It was not until fast, relatively high-performance microprocessors became available for personal computers, giving them the power to run concurrent tasks satisfactorily, that multitasking became a popular word in the software development field.

PC-DOS does not presently have a multitasking version available. There are, however, several software products that attempt to add multitasking to the existing versions of PC-DOS, with varying degrees of success. One such product is MultiLink from The Software Link, Inc. (not to be confused with Davong's MultiLink local area network). This review will discuss version 2.07 of the MultiLink package.

MultiLink claims to be not only a multitasking system, but also a multi-user system. The two concepts are distinct, and in evaluating such products it is important to realize the difference. Computer operating systems can be roughly divided into three categories: (1) PC-DOS alone is a single-tasking, single-user operating system in which only one person can use the PC, and only one program can be run at a given time; (2) most older mainframe time-sharing systems are multi-user systems, which, from the viewpoint of a single user, *appear* to be single-tasking systems because a user can run only one program at a time from his terminal. In reality, however, such a system is usually multitasking because it is running many users concurrently; (3) newer systems, such as UNIX, allow multiple users and allow each user to run more than one task from his terminal.

MultiLink is in the second of these categories, having reached what might be considered the second step in the evolution of multitasking systems. However, it provides an extension to the multi-user, single-task concept that allows someone at the PC's keyboard to switch among several tasks. This capability is different from that of systems like UNIX, which has a more generalized multitasking capability.



J. Eric Roskos is a graduate student in computer science at Vanderbilt University. His dissertation is on a new architecture for shared memory in multiprocessor systems.

USING MULTILINK

The MultiLink program is started in much the same way as the PRINT command of DOS 2.0, which also attempts to provide limited multitasking. Fortunately, MultiLink's operation is much less erratic.

Typing the command MLINK (along with a string of configuration parameters) causes MultiLink to begin executing. A copyright notice is displayed on the screen, but then the program appears to exit, returning the user to the command prompt. During the brief interval that the program runs, however, it is installing itself in memory as an extension of DOS and will remain there until the system is rebooted.

After MultiLink has been started, nothing at first seems different from regular PC-DOS. An A> prompt still is displayed on the screen, and the system still accepts commands as usual. However, the ability to switch to one or more "background partitions" is now available; this is done by pressing ALT-F*n*, where *n* is the number of the partition that is to be accessed. Each partition has its own command prompt: thus the user can switch to a background partition; start a compiler, print-formatting program, or other program requiring little keyboard interaction; and then switch back to the main (foreground) partition and continue to work while the other program is running. Alternately, if a conventional terminal is available, it may be plugged into the PC's COM port, and a second user can run programs there at the same time the first user is running programs from the PC's keyboard.

At the time MultiLink is started, the user must specify the size of one or more *partitions* in which tasks may run. The size and number of partitions may not be altered once MultiLink has started. This concept of "memory partitioning" first appeared many years ago in mainframe multitasking systems, and it represents an evolutionary step that precedes the current state-of-the-art. Nevertheless, with proper planning memory partitioning can be used with no problems, although it is not as flexible as the more recently developed dynamic memory allocation schemes used by UNIX and similar systems.

In memory partitioning, if the user allocates one 192K and one 64K partition on a 256K system, he will not be able to run two 90K programs at the same time, even

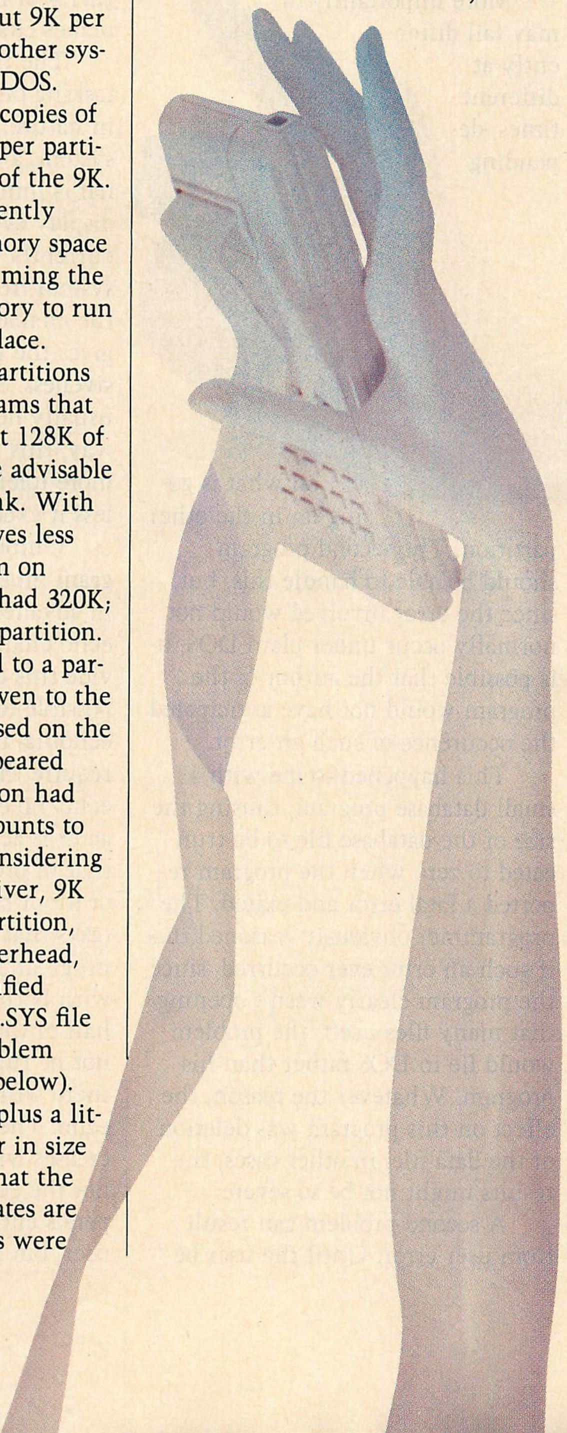
though there would be 76K of memory left over if both were running. This is because MultiLink creates the illusion that two PCs are being used—one with 64K and one with 192K; and clearly a 64K machine could not run a 90K program.

The fact that some memory must be used by both DOS and MultiLink must also be taken into account. Unlike some of the more primitive multitasking systems for the PC, MultiLink requires only one copy of DOS to be loaded into memory. This saves some space. MultiLink takes up about 13K for its own software, plus about 9K per partition "for buffers and other system areas," in addition to DOS. MultiLink loads multiple copies of COMMAND.COM—one per partition—accounting for part of the 9K. However, the 9K is sufficiently small that no serious memory space limitations are likely, assuming the system has adequate memory to run multitasking in the first place.

Because of the fixed partitions and the size of most programs that run under PC-DOS, at least 128K of memory would seem to be advisable for a PC running MultiLink. With just two partitions, this gives less than 64K each. The system on which I tested MultiLink had 320K; it was run with one 128K partition. Any memory not allocated to a partition at start-up time is given to the "foreground partition"; based on the CHKDSK command, it appeared that the foreground partition had 141K available, which amounts to about 50K of overhead. Considering DOS (24K), a 3K device driver, 9K for the one background partition, and the 13K MultiLink overhead, this adds up to 49K. I specified FILES=16 in my CONFIG.SYS file in order to overcome a problem with file usage (described below). The 312 bytes this added, plus a little room for rounding error in size estimates, seems to show that the MultiLink manual's estimates are correct. (These calculations were done under version 2.06. Version 2.07 adds 3K to the size of the resident portion of MultiLink as a result of program enhancements.)

How It Works

MultiLink works by installing itself between the operating system and the programs that run on the PC. Whenever a program calls up DOS (or in some cases the ROM BIOS), the program must first go through MultiLink, although this process is invisible to the program. MultiLink can then decide that it's time to run one of the other concurrently active programs, and it gives control of the system to the new program instead of continuing the original program that made the call to DOS. According to the manual, it does this after 1/18 second has elapsed. In fact, a considerably longer time, possibly even several minutes, could pass in the case of a program that rarely calls DOS, were it not for an optional feature known as *time-slicing*.



Time-slicing assures that a program in a multitasking environment gets its fair share of time at sufficiently regular intervals. This time situation is one that is often overlooked in simplified descriptions of multitasking, which say that the system works by running a second task while the first is waiting for some event such as I/O. In reality, a multitasking system that worked only in this way would behave very erratically. In many cases, the waiting time would be far too brief to allow two or more concurrent tasks to make reasonable progress.

In time-slicing, the multitasking system keeps track of how long the currently running task has been executing. If the task does not voluntarily allow other tasks to run after a certain amount of time has elapsed by waiting for I/O, the system stops the task anyway and allows other tasks to run for an equal time. The amount of time each task is allowed to run is called the *quantum* and is typically between 1/20 of a second and 1 second.

To use MultiLink's time-slicing option, the command MLSLICE must be given before MultiLink is started. Without the command, MultiLink will switch between tasks only when DOS is called. This involves less byte overhead, but can make operation erratic when running programs that call DOS rarely.

How Well Does It Work?

Considering that DOS was not designed to support multitasking, MultiLink does a surprisingly good job. It is not perfect, however. Both

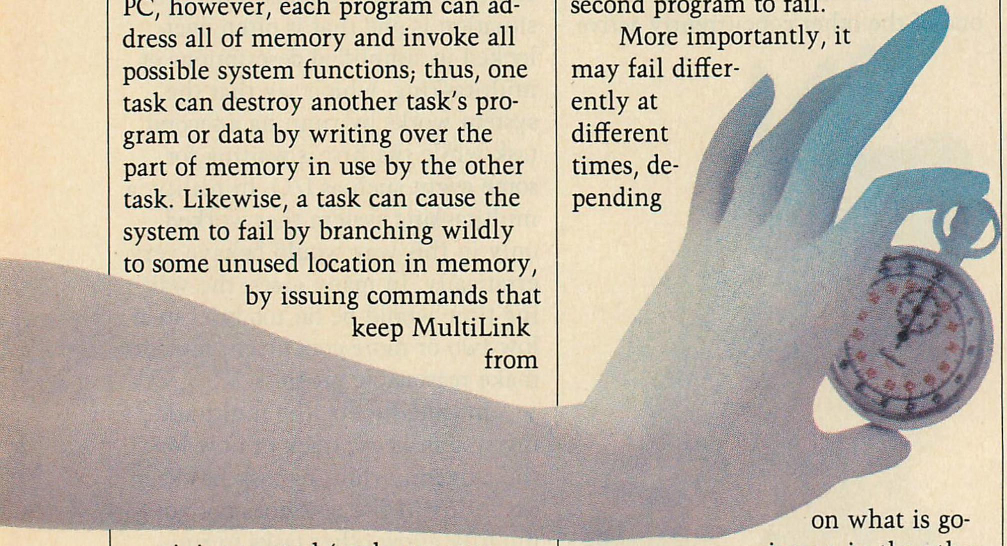
DOS and the hardware have some limitations that noticeably affect performance. Furthermore, MultiLink is still growing and thus currently lacks some features that would be useful.

The IBM PC was not designed to support multitasking. Machines designed for such a capability usually contain features that insure a program running on the system cannot interfere with the other concurrently running programs, either by accident or intentionally. On the PC, however, each program can address all of memory and invoke all possible system functions; thus, one task can destroy another task's program or data by writing over the part of memory in use by the other task. Likewise, a task can cause the system to fail by branching wildly to some unused location in memory, by issuing commands that keep MultiLink from regaining control (such as a command that turns the clock), or even by invoking the reboot function provided in the ROM BIOS.

It is worth noting that some of these problems can be avoided on the PC if each user is content with a maximum of 128K. The often-criticized "small memory model" provided by many compilers can prevent, at least, overwriting of data in memory, if each program is placed outside the 128K range that the other programs are able to address. In fact, the small memory model also provides some other multitasking advantages that are beyond the scope of this article; but it does not prevent all the ways in which a program can cause the system to fail.

Another problem involved in multitasking has to do with DOS itself. DOS is a single-user operating system; it doesn't know about MultiLink and functions as if only one program were running at a given time. Thus, three problems can often occur when the system is being used. Two of these problems can be catastrophic.

First, if one of two progams under MultiLink opens a large number of files using the DOS 2.0 file system functions, then the maximum number of files that may be simultaneously opened may be exceeded when the second program tries to open *its* files. This may cause the second program to fail.



More importantly, it may fail differently at different times, depending on what is going on in the other partition. The second program should be able to handle this; but, since the error involved would not normally occur under plain DOS, it is possible that the author of the program would not have anticipated the occurence of such an error.

This happened to me with a small database program, causing the size of the database file to be truncated to zero when the program reported a fatal error and exited. The programmer obviously reasoned that if such an error ever occurred, since the program clearly wasn't opening that many files itself, the problem would lie in DOS rather than his program. Whatever the reason, the effect on this program was deletion of the data file; in other cases, the results might not be so severe.

A second problem can result from user error. Until the user becomes accustomed to running MultiLink, he may forget that another program is quietly running in a background partition. If he were to change disks at this point he would destroy the running disk's file allocation table, causing all files on the disk to become inaccessible. (In some cases, the RECOVER command will get them back, although this is very time consuming.) For this reason, MultiLink is probably better suited for systems with a hard disk than for those with only floppy disks, because the limited capacity of floppy disks sometimes makes changing them a necessity.

The third limitation of multitasking on a PC is purely aesthetic in nature. On a large time-sharing system, a character typed at a terminal is immediately echoed on the display even if another task is concurrently running, preventing the system from doing anything with the character at that time. This gives the illusion of better responsiveness. Since some waiting is usually needed in such a case anyway, this form of delay is usually more tolerable than echo delays, unless it exceeds several seconds.

Unfortunately, to do this, a program must tell the operating system in advance whether or not it should echo characters. DOS does not provide this capability. Although it is possible to make a good guess about echoing, ultimately DOS does not require the program to specify the echo/no echo attribute until a character is actually read. This means that in order for echoing of a string of input characters to occur, the tasks that use them must be running. Since, in a two-task system with both tasks attempting to run, half of the time a given task will *not* be running, half of the time there will not be an immediate echo. The probability of no echo increases with more tasks. Whether or not the echo delay is acceptable depends entirely on the individual user; but it makes performance

Debugging MultiLink

Perhaps the most serious problem discovered during the review of MultiLink was interference with the prototype board, which required a hardware modification to the PC.

Since the problem involved MultiLink's writing to an I/O port somewhere within the 13K program, and since MultiLink does a lot of writing to I/O ports throughout, using the debugger to search for I/O instructions would have taken a long time and would have been inexact.

A logic analyzer could be connected to the CPU to watch for I/O to the port in question. But since I don't have access to a logic analyzer, I made the PC serve that function.

To do this, I connected pin 8 of U4 on the IBM prototype board to pin B4 of the expansion slot into which the prototype card was plugged. Pin 8 of U4 generates a logic "high" pulse whenever the prototype board's I/O ports are written to; and pin B4 of the expansion slot is the interrupt request line for interrupt type 2. When pin B4 gets a logic "high" pulse, it generates an interrupt of type 2, causing a branch to the interrupt vector at address 28H in RAM. Thus, the hardware modification would cause an interrupt whenever MultiLink wrote to the prototype board.

Next, I wrote a simple program (shown in listing 1) that installed itself in much the same way MultiLink does, establishing itself in memory and pointing the interrupt vector at 28H to a routine within the program. This routine would display the address at which the interrupt occurred (which was stored on the stack by the 8088 when the interrupt occurred), then return. The net effect was that, whenever the prototype board was addressed, the address of an instruction immediately following the instruction that wrote to the prototype board would be displayed on the monitor, but the program would continue execution.

This quickly pinpointed the problem areas, shown in figures 1 and 2. (In these displays, 100H must be added to any addresses in the instructions. The debugger shows offsets relative to the start of the program segment prefix, but MultiLink was assembled without the ORG 100H instruction that accommodates for the PSP in .COM programs.)

During start-up, as shown in figure 1, MultiLink clearly writes to the prototype board at I/O address 300H. It does this at address 327F in the debugger display; it loads the DX register with the word at location 1FF4 (20F4 after adding 100H to it), which contains the number 300H. It then outputs the value in the AL register to that address, which is the first address on the prototype board.

This, in fact, appears to happen inside a loop that initializes the partition tables, because right after it does the OUT instruction, it subtracts B0H from the SI register, and if the result is not zero, it branches back to an address earlier in the program. Since B0H is the size of one entry in the partition table, according to the MultiLink manual, the initialization loop appears to be for the partitions. This was supported by the I/O tracer, which showed that the prototype board was accessed twice in rapid succession, with the background partitions set up at the time.

Although this first case clearly seems to be an intentional write to the prototype board, the second case (shown in figure 2), which occurs when the function keys are depressed, could simply be a programming error. Again, an OUT instruction occurs at address 1209H. But the DX register, which points to the I/O port, is initialized long before this; two subroutine calls (to save or restore the display memory when switching partitions) take place in the meantime.

Although it is not shown in listing 1, the DX register is loaded in a very roundabout way—by POPping its value, which was PUSHed there even earlier in the program, off the stack. Since the write to the prototype board occurs only occasionally and involves seemingly random values, this example may be an error in the program resulting from the roundabout way in which the DX register is set up.

To eliminate the interference with the prototype board, both OUT instructions were changed to NOP instructions. Although it is not clear exactly what the instructions did, MultiLink's single-user mode works properly, and MultiLink no longer interferes with my prototype board.

—J. Eric Roskos

```
365F:3290 E966FE        JMP     30F9
365F:3293 1E            PUSH    DS
365F:3294 8E1E6C00      MOV     DS,[006C]
365F:3298 BA8000        MOV     DX,0080
365F:329B B41A          MOV     AH,1A
365F:329D CD21          INT     21
-e 20f4
365F:20F4  00.     03.
-u 3279
365F:3279 8B16F41F      MOV     DX,[1FF4]
365F:327D 42            INC     DX
365F:327E FA            CLI
365F:327F EE            OUT     DX,AL
365F:3280 FB            STI
365F:3281 B019          MOV     AL,19
365F:3283 B500          MOV     CH,00
365F:3285 B618          MOV     DH,18
365F:3287 E8F9F2        CALL    2583
365F:328A 81EEB000      SUB     SI,00B0
365F:328E 7E03          JLE     3293
365F:3290 E966FE        JMP     30F9
365F:3293 1E            PUSH    DS
365F:3294 8E1E6C00      MOV     DS,[006C]
365F:3298 BA8000        MOV     DX,0080
-
```

Figure 1: First Debugger Display

```
-
-
-
-
-
-
-
-
-
-
-u 11f0
35D2:11F0 26            ES:
35D2:11F1 8A3E6204      MOV     BH,[0462]
35D2:11F5 B402          MOV     AH,02
35D2:11F7 E8BE0C        CALL    1EB8
35D2:11FA 8E063C09      MOV     ES,[093C]
35D2:11FE 33FF          XOR     DI,DI
35D2:1200 BE2A25        MOV     SI,252A
35D2:1203 B9A00F        MOV     CX,0FA0
35D2:1206 E8B600        CALL    12BF
35D2:1209 EE            OUT     DX,AL
35D2:120A 5E            POP     SI
35D2:120B E98500        JMP     1293
35D2:120E 39065A00      CMP     [005A],AX
-
```

Figure 2: Second Debugger Display

seem worse than it is, and it makes it take longer for a fast typist to feel comfortable with the system.

Despite these limitations, MultiLink has some powerful features. Version 2.07 supports a wide variety of terminals for those who want to use their machines as multi-user systems. A program that writes only to the display through DOS or BIOS calls will work very well on a remote terminal under MultiLink. A program that writes directly to the screen memory, however, may or may not work.

MultiLink provides a program called MLVIDRAM that is supposed to help in this case. The version 2.06 manual claimed that MLVIDRAM is documented separately for those familiar with assembly-language programming, but no documentation on how to use this program was provided except for a set of the more popular applications programs. The 2.07 manual deletes this reference altogether. The program MLVIDRAM works for many popular applications programs, however, and the company seems willing to enhance it for any programs that are not supported.

MLVIDRAM is used in conjunction with a program called MLVIDFIX. A batch file must be created to run MLVIDFIX, which patches the applications program before each execution in order for it to run in a particular partition. Because of the way MLVIDFIX works, a slight possibility exists that the program could fail if two users tried to start the same application at the same time (since both users would be trying to apply different patches to the same program).

PROGRAMMING INTERFACE

MultiLink provides a number of advanced functions via interrupt 7FH, which is invoked in the same way as the DOS interrupt 21H functions. The advanced functions include an implementation of the enqueue/dequeue synchronization primitives; the ability to read and set some—but not all—of the serial port's control/status bits; an unusually straightforward way to set the baud rate for the serial ports; the ability to change the terminal type for a remote port on a multi-user configuration; and several features that improve system performance. Some of the latter features, particularly one intended to suspend a task if it is not being useful, seem to have been provided to overcome limitations in compiled BASIC and don't always work with other languages.

This is the most impressive feature of the MultiLink system: an elegantly implemented spooler that is simple and yet effective.

The advanced functions are further enhanced by a detailed description of MultiLink's internal status table; some functions that are not available through interrupt 7FH may be simulated by modifying this status table appropriately.

A few useful functions are missing at present. The ability to specify echoing of characters to overcome the echo delay problem would be helpful. Another valuable addition would be the ability to take control of a serial port that is normally assigned to a remote partition for use by another task; this would be useful, for example, for the user who has a modem connected to his serial port so people can dial up the modem and use the system, but who also wants to be able to run a terminal emulator using the modem from time to time. (MultiLink provides a way to do this if the serial port is not assigned to a partition. I found, by trial and error, another way to accomplish this by using modifications to the system status table.)

A generalized event wait function, which would cause a task to cease running until some external event occurred, would be helpful in improving *MultiLink's* multitasking performance. This is partially implemented now, but it is difficult to use effectively. The company claimed at the time version 2.06 was released that a generalized event wait would be added in the future, but it was not added to version 2.07.

A priority-scheduling function is also available; but the method presently used is of limited utility in an interactive system, since it allows a higher-priority task to "lock out" a lower-priority one, rather than simply scheduling higher-priority tasks more frequently than lower-priority ones.

UTILITY PROGRAMS

Several utility programs are provided with MultiLink. The best of these is a new print spooler provided with release 2.07. This is by far the best print spooler for the IBM PC; its operation is similar to that of mainframe print spoolers. When the spooler is activated, all output to the printer (regardless of whether it was output through BIOS calls or through the device PRN) is sent to a disk file. The disk files are normally closed when a program exits, so that a separate file is created for each run of a program. A separate print program, which runs in an 18K partition, scans a user-specified directory every few seconds for a new file to print.

The spooler allows the user to specify disposition (Keep, Delete, etc.), priority, and class (used to designate special forms) for each print job. A continuously updated display is kept in the print program's partition, showing what the spooler is currently doing and enabling the user to give spooler commands. This is the most impressive feature of the MultiLink system: an elegantly implemented spooler that is simple and yet effective.

In addition to the spooler, a bulletin board system, MLBBS, is provided. Although not as flexible as some bulletin board systems, it is a pleasing program to use. The source code, which is included, is written in a dialect of BASIC that allows named line labels instead of line numbers. A preprocessor converts this file to a standard BASIC file.

A few support programs are also part of the MultiLink package. One such program is a "keyboard enhancer" that functions similarly to the ANSI.SYS device driver's function-key-assignment feature: any key can be translated to an arbitrary string. Furthermore, the input to the translation process can be a short string, such as the escape sequences sent by many terminals, so that a terminal's function keys can be assigned to strings at the remote ports. The program that is used to make these assignments is highly interactive. Since MultiLink also provides a set of escape sequences to simulate the PC's function keys, the user is able to translate from the terminal's function keys to the PC's function keys, if those keys are available on the terminal.

A general parameter-setting program is also provided; this program allows the user to set the terminal type, baud rate, priority, and other attributes of a partition.

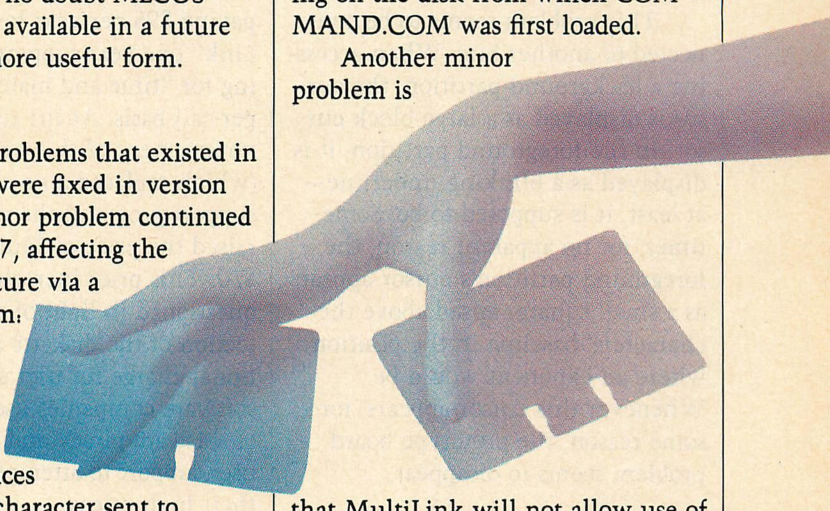
One utility that comes with release 2.07 of MultiLink is amusing, although it could be frustrating as well. According to the manual, if the user has a terminal not currently supported by MultiLink, he can specify terminal type 9 and define this terminal to MultiLink via the program MLCUSTOM. This program, the manual says, also lets the program be configured for nonstandard serial cards. A sample run of the program came up with the following message:

> Call us at (404) 255-4210 if you need help with installation of serial ports or terminal type 9.

This may be merely a case of the manual being ahead of the software development; no doubt MLCUSTOM will be available in a future release in a more useful form.

Problems

Most of the problems that existed in version 2.06 were fixed in version 2.07. One minor problem continued in version 2.07, affecting the multi-use feature via a dial-up modem: the program seems to send cursor-positioning escape sequences before every character sent to the terminal, even when they are unnecessary, so that the actual rate at which characters appear on the terminal's screen is noticeably slower than would otherwise be the case. This made output "sluggish."



One other minor problem is that MultiLink will not allow use of the DOS ASSIGN command prior to start-up. If MultiLink detects that an ASSIGN command has been used, it outputs an error message and aborts start-up.

Additionally, version 2.07 does not take advantage of the COMMAND.COM feature (under DOS 2.0) that allows the user to specify the directory in which COMMAND.COM can find its nonresident segment. When MultiLink was tested in conjunction with Turbo Pascal, the system came to a complete halt in all partitions as soon as the user running the compiler exited Turbo. COMMAND.COM tried to reload its transient portion and looked for the file on the currently logged disk, rather than looking on the disk from which COMMAND.COM was first loaded.

Another minor problem is that MultiLink will not allow use of the DOS ASSIGN command prior to start-up. If MultiLink detects that an ASSIGN command has been used, it outputs an error message and aborts start-up. This is not too serious, but it requires a separate AUTOEXEC.BAT file for each partition if device letter assignments must be made. Then the user will be able to give the ASSIGN commands after MultiLink has started.

Although version 2.07 fixed many of the minor problems of 2.06, one very irritating and serious problem was actually made worse in the new release. In version 2.06, MultiLink would output a byte to the IBM prototype board during its initial start-up. On a system that has some sound-producing devices on the prototype board, this can be quite bothersome, and requires writing a program to be run by the boot disk's AUTOEXEC.BAT file in order to turn the devices off. This results in loud noises coming from the prototype board whenever the system is booted. The situation could be more serious for those who use the prototype board to interface to laboratory equipment.

In release 2.07, data are written to the prototype board not only when the system is started up but also almost every time the function keys are used to switch between

partitions. Furthermore, the data being output seem more random than in version 2.06.

This problem seems to be connected to another bug. When accessing a background partition, the cursor is displayed as a large block cursor. In the foreground partition, it is displayed as a blinking underline—at least, it is supposed to be. Sometimes, for no apparent reason, the foreground partition's cursor appears as a small square, raised above the characters' baseline in the position where an exponent would be. Whenever this cursor appears, for some reason, the prototype board problem seems to disappear.

Documentation, User Support, and Price

Documentation consists of a 61-page booklet the same size as the standard IBM documents. This is a considerable improvement over the version 2.06 manual, which was much shorter. The documentation describing the programming interface has not been significantly expanded from the previous manual; but the level of detail is comparable to that of the DOS 2.0 manual's documentation of system calls.

In the realm of user support, the Software Link provides a dial-up bulletin board for MultiLink users. The bulletin board provides up-to-date copies of the documentation, as well as a few utility and game programs, and occasional information on other programs.

In mid-March, the Software Link announced a change in its user support policy: it now provides only 30 days of free user support; thereafter, the user must choose between paying $95 per year for the "Service Link" one-year support plan or paying for "time and materials" on a per-call basis. At the time of the announcement of this new policy (which replaced a no-charge user support program) the company raised the price of MultiLink by $70. This price increase could be questioned in light of the copy-protection of the diskette and the additional charge for user support; many software companies use the problem of software piracy and the cost of user-support to attempt to justify their high prices.

MultiLink's version 2.07 now costs $295, although a demo version is available for $25. With patience, the user can give the program a thorough workout using the demo version. It is identical to the standard system, except that it quits scheduling tasks to run after 7 minutes, effectively halting the system. The program is copy-protected, requiring the original program disk to be inserted when the system is first started. The program disk may be reformatted (but not bulk-erased) without destroying the copy-protection function. The protection scheme is described as a "fingerprint" method, although it is simpler than the Prolok scheme usually described by this term.

Although MultiLink does not represent the state-of-the-art in multitasking software in comparison to other operating systems available for the IBM PC, it does a reasonably good job and is certainly a good choice for those needing multitasking under PC-DOS.

The product's high price, compounded by the optional service contract, places it at the upper boundary of price/performance.

Several other companies have already produced multitasking operating systems for the IBM PC, including both Concurrent CP/M-86 and PCIX (IBM's own adaptation of Bell Labs' UNIX Time-sharing System), and it is likely more and more multitasking software will emerge. It is possible that a future release of PC-DOS could include multitasking. Such an enhancement to DOS would most likely make multitasking add-ons obsolete, unless the new DOS implementation had fairly serious performance problems.

For now, MultiLink provides a very good way to enjoy multitasking advantages while still having access to the large body of software that runs under PC-DOS.

MultiLink
The Software Link, Inc.
6700 23-B Roswell Road
Atlanta, GA 30328
404-998-0700
$295
CIRCLE 490 ON READER SERVICE CARD

Listing 1 INTRACE Display Whenever an Interrupt Type 2 Occurs

```
	page	56,132
	title	INTRACE Display whenever an interrupt type 2 occurs
	.sall
;
; Program to display the interrupted address whenever an interrupt
; of type 2 occurs.  This is used to find out where MultiLink is writing
; to the IBM Prototype Board.
;
; If this program seems to have some superfluous code, it's because it
; was originally used in tracing disk I/O calls to debug a device
; driver...
;

;
; Macros
;

;
; wto: write message to display.  Append cr/lf unless crsup=nocr
```

```
;
wtomacro  msg,crsup
   local  msgstr,around

   jmp    around

msgstr    db      msg
   ifb    <crsup>
   db     0DH,0AH
   endif
   db     '$'

around:   push    ax
   push   bx
   push   si
   push   di
   push   bp

   mov    si,offset msgstr
   call   putc

   pop    bp
   pop    di
   pop    si
   pop    bx
   pop    ax

   endm

prtreg    macro   rg,msg  ; put register rg with message msg on display

   push   ax

   mov    ax,rg
   call   prtax
   wto    msg,nocr

   pop    ax
   endm

;
; Code Segment
;

cseg              segment para public 'code'
          assume  cs:cseg,ds:cseg,ss:cseg,es:nothing

          org     100H
;
; COM program startup
;
cpstart           proc    far
          jmp     near ptr start
cpstart           endp

;
; Local Procedures
;
putc              proc    near    ; write a string to display W/O DOS
                                  ; intervention
          mov     bl,7    ; Parameter setup same as for DOS's
                          ; write-string
          mov     bh,0    ; function
putc1:
          mov     al,cs:[si]
          cmp     al,'$'
          je      putc2
          mov     ah,14
          int     10H
          inc     si
          jmp     putc1

putc2:
          ret
putc              endp
```

```
prtnum            proc    near    ; print half a byte on screen
            push  ds      ; this procedure is used by prtax below
            push  cs
            pop   ds
            push  bx
            mov   bx,offset xltab
            xlatb
            mov   ah,14
            mov   bh,0
            int   10h
            pop   bx
            pop   ds
            ret
xltab             db      '0123456789ABCDEF'
prtnum            endp

prtax             proc    near    ; print contents of ax register
                                  ; on screen
            push  cx      ; all registers are preserved
            push  ax
            mov   al,ah
            mov   cl,4
            shr   al,cl
            call  prtnum
            pop   ax
            push  ax
            mov   al,ah
            and   al,0Fh
            call  prtnum
            pop   ax
            push  ax
            mov   cl,4
            shr   al,cl
            call  prtnum
            pop   ax
            push  ax
            and   al,0Fh
            call  prtnum
            pop   ax
            pop   cx
            ret
prtax             endp

;
; this routine does the actual display of the interrupted location
;

sssav             dw      ?
spsav             dw      ?
axsav             dw      ?

lsbottom  db      1024 dup (?)
lstack            dw      0

tracer            proc    near

            ;
            ; establish our environment
            ; we turn off ints here for old 8088s with defective
                ; microcode
            ; (e.g., mine)
            ;

            cli
            mov   cs:axsav,ax               ; switch stacks
            mov   cs:sssav,ss
            mov   cs:spsav,sp
            mov   ax,cs
            mov   ss,ax
            mov   sp,offset lstack
            sti

            mov   ax,cs:axsav               ; get ax back

            ;
            ; print address to which the RTI will return
            ;
```

```
        push    ax
        push    bx
        push    es

        mov     es,cs:sssav
        mov     bx,cs:spsav

        wto     "int2: ",nocr
        prtreg  es:2[bx],":"
        prtreg  es:[bx]," "
        wto     " "

        pop     es
        pop     bx
        pop     ax

        mov     al,20H          ; signal EOI
        out     20H,al
        mov     ax,cs:axsav

;
; reestablish original environment
;

        cli                             ; restore stack
        mov     ss,cs:sssav
        mov     sp,cs:spsav
        sti

        iret

tracer          endp

;
```

```
; everything below this point will be deleted from memory once the
; initial command exits
;

start           proc    near

        wto     "Installing interrupt trace."
        mov     ax,0
        mov     es,ax

        mov     ax,offset tracer
        mov     word ptr es:28H,ax
        mov     ax,cs
        mov     word ptr es:2aH,ax

        wto     "Enabling interrupt"

        mov     al,0B8H
        out     21H,al
        mov     al,20H
        out     20H,al

        wto     "Interrupt enabled"

        wto     "Installation complete; executing TSR."
        mov     dx,offset start
        add     dx,100H ; a little extra space...
        int     27H

start           endp

cseg            ends

        end     cpstart
```

We just made IBM's PC/XT 20 times more powerful.

Introducing Dragon's New 65 Mb and 140 Mb Hard Disk Drives.

$4995

Now available for immediate delivery.

$6995

What makes Dragon the best hard disk drive for the IBM PC, XT, and compatibles?

Larger Storage Capacity...

With 65 Mb of on line storage, the SP-65 holds up to 57 Mb of formatted data. The SP-140 stores up to 121 Mb of formatted data with a total capacity of 140 Mb. No 5¼ inch Winchester stores more than these new Dragon Disks.

Faster Access Time...

At 30 msec. average access time, the SP-65 and SP-140 are faster than other 5¼ inch Winchesters... 3 to 4 times faster.

Better Features...

The SP-65 and SP-140 come with hard disk, controller, host adapter, power supply, cables, and chassis. Both systems are compatible with PC-DOS and MS-DOS.

More Options...

Additional options include: Dual drives with up to 280 Mb of on line storage (or 260 Mb formatted). Streamer tape and random access back up. 90 day warranty. Responsive technical help.

For more information, contact Dragon Industries, 35 Main Street, Hopkinton, Mass., 01748, (800) 824-8006 or (617) 435-4183 in Massachusetts. Distributor and OEM inquiries invited.

Dragon Industries

Turning Micro Computers into Monster Computers.

CIRCLE NO. 144 ON READER SERVICE CARD

Dragon Industries
35 Main Street, Hopkinton, Mass. 01748

Name ____________________

Company ____________________

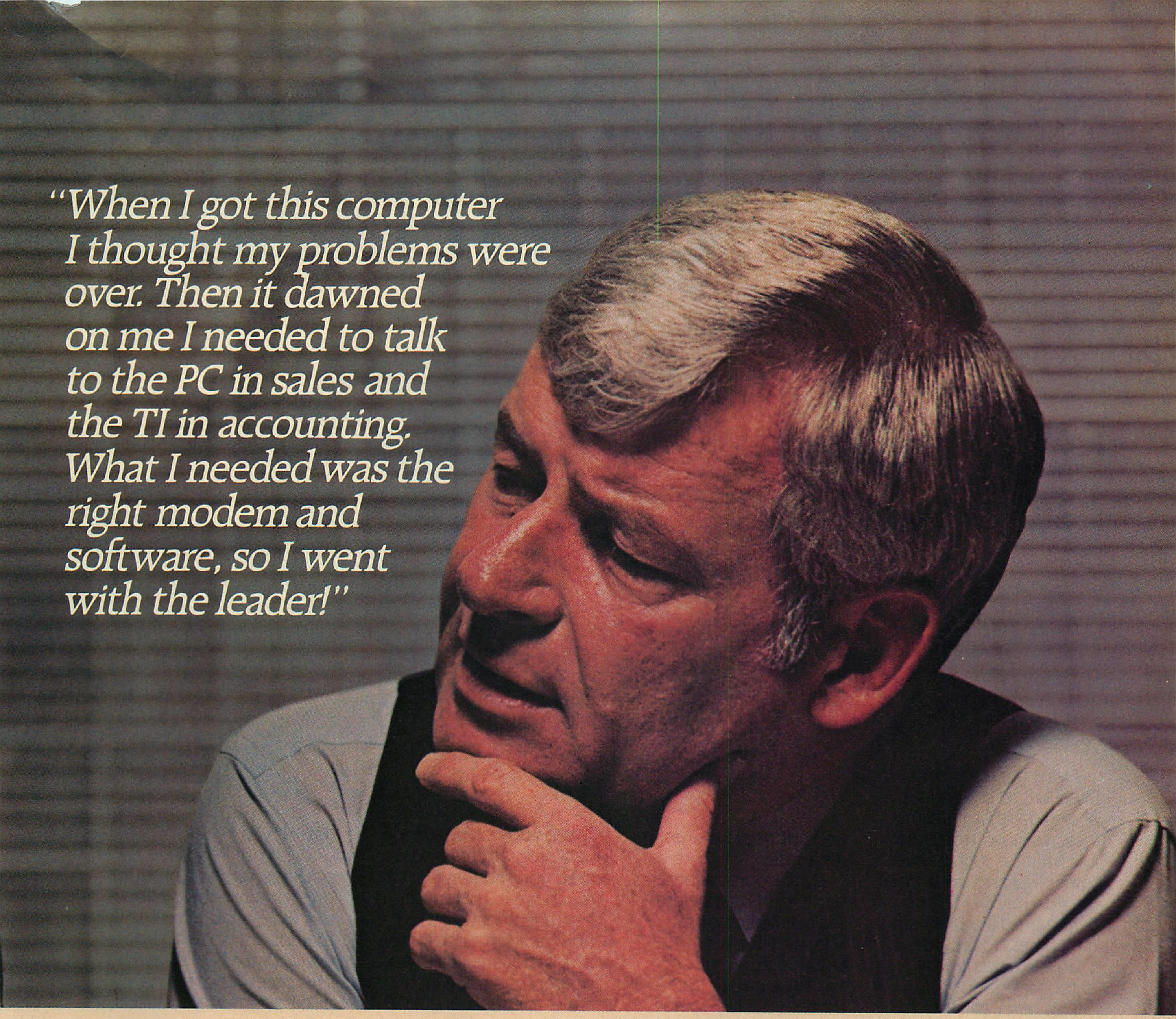
Address ____________________

Telephone ____________________

Computer ____________________

OS Version ____________________

Interested in: ☐ SP-65
☐ SP-140



Voice to data—in the same call!

With Smartcom II, you can easily switch from voice to data transmission (and back again), all in the same phone call. This saves you time and money, since you don't have to hang up and dial again.

Your Hayes telecomputing system works—totally unattended.

Smartcom II makes telecomputing simple, even when you're not there. It allows your Smartmodem to receive a message for you when you're out, and leave it on your disk or printer. And you can tell Smartcom II to "save" the messages you've created during the day, and automatically send them at night, when phone rates are lowest.

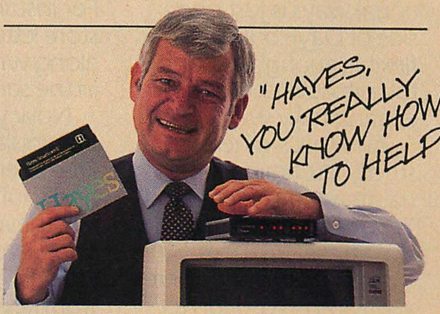
Get your hands on the leader.

With an unsurpassed record of reliability, it's a small wonder Smartmodem is such a smart buy! Smartmodem 300™ (the first of the Smartmodem series) dials, answers and disconnects calls automatically. Smartmodem 1200™ and Smartmodem 1200B™ (it plugs into an expansion slot inside an IBM PC or compatible), provide high-speed, high-performance communications for businesses of all sizes.

And when Smartmodem is purchased with Smartcom II, you have the most dependable telecomputing system available for your personal computer.

Everything we do at Hayes is designed to make communications easier for you. Feature-rich, direct-connect modems. Menu-driven software. Concise documentation. *And* a customer service organization, second to none!

See your dealer right now for a hands-on demonstration of Smartmodem and our latest version of Smartcom II. From the telecomputing leader. Hayes.

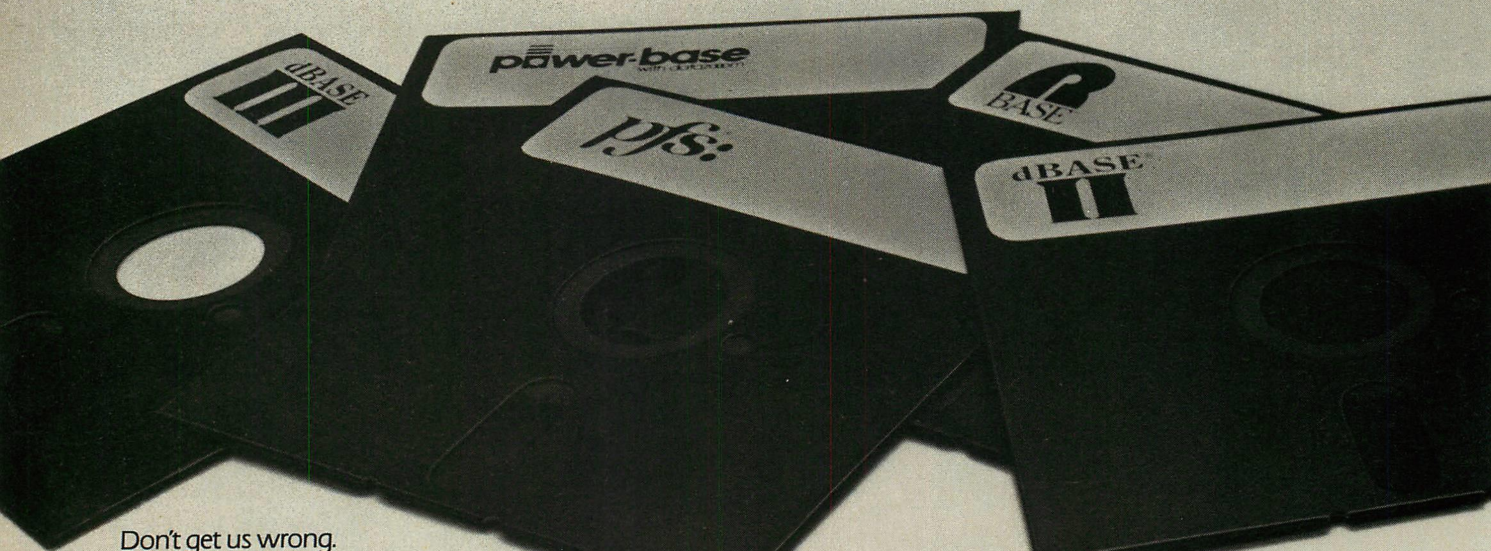


Hayes™

Hayes Microcomputer Products, Inc.
5923 Peachtree Industrial Blvd.
Norcross, Georgia 30092 404/441-1617.

CIRCLE NO. 158 ON READER SERVICE CARD

Traditionally, databases have been left-brained.



Don't get us wrong. DayFlo is as left-brained as any database around. It can handle structured information such as customer lists and personnel records. Keywords and field names are used to gain access to records.

Just like traditional databases.

However, solving traditional problems in traditional ways is just one side of DayFlo. It's also flexible enough to handle today's new challenges in exciting new ways.

And that's where strictly left-brained databases get left behind.

The First Free-Form Database.

In the real world, there's always an exception to the rule.

Which wreaks havoc on traditional databases. For example, because of fixed formats, entering a new field such as a Telex number into a single customer record is next to impossible. Unless you're prepared to reprogram and reformat your entire database.

DayFlo, on the other hand, eliminates these tedious operations.

Word-Oriented Database.

Unlike traditional databases, DayFlo has integral word processing. For instance, you can store letters and memos along with customer lists in the same database.

Now you're free to manipulate data at will. Free to enter, retrieve, edit and shuffle information whenever and wherever you please.

Every fact you need, from key client lists to spontaneously entered notes and ideas, is just a few keystrokes away.

Power, Yes. Programming, No.

DayFlo is a powerful package that runs on an IBM PC with hard disk, Compaq Plus, or compatibles.

Yet, you don't have to be a programmer to use it.

So you see, DayFlo is a much greater information management tool than any traditional database. Much more flexibility and power than pfs. Much easier to use than dBASE II.

ReportFlo.

With our ReportFlo package, you can produce presentation-quality documents, reports, letters, memos and so forth from the data stored in DayFlo. This potent report writer also performs calculations.

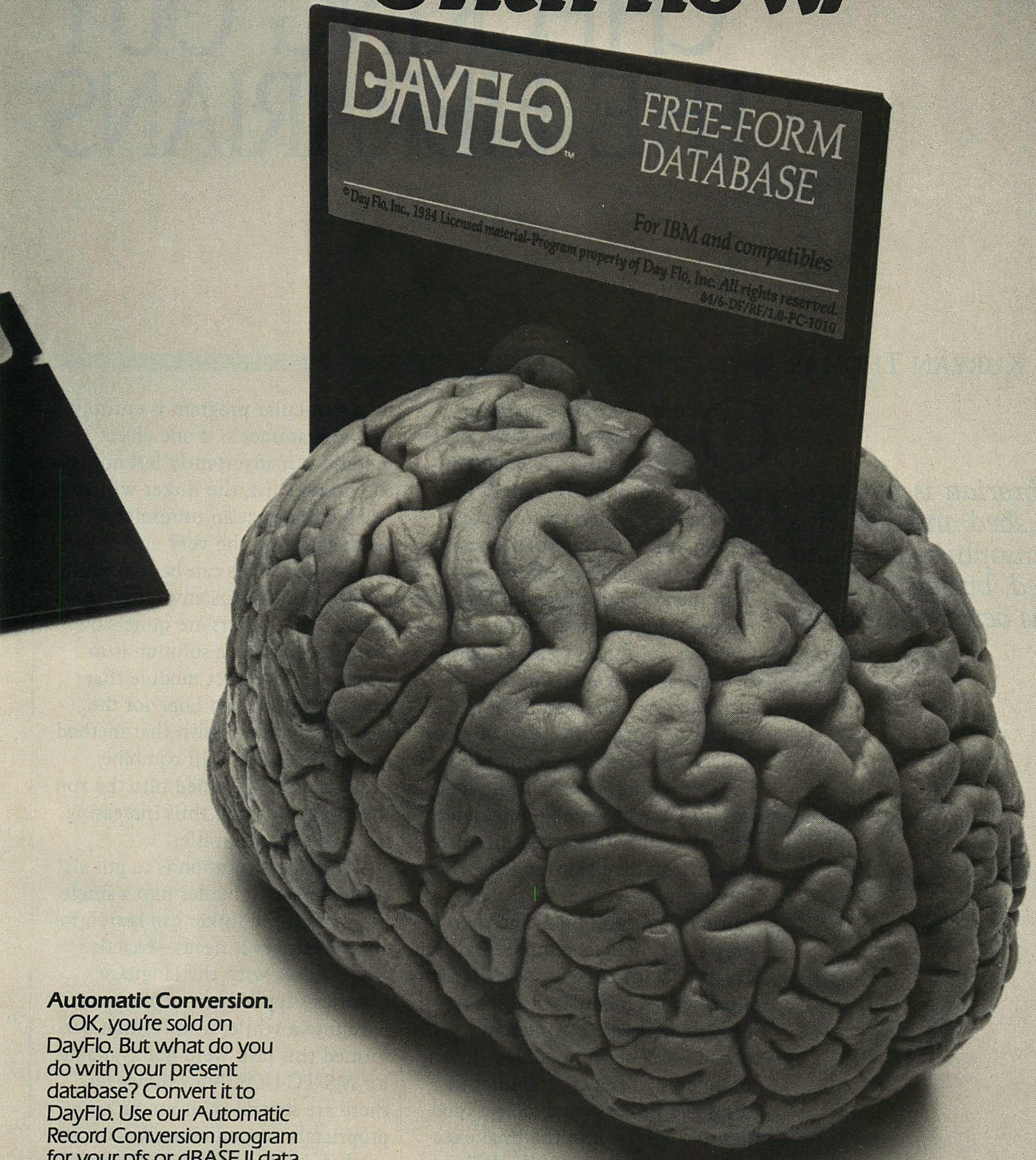
COMPARING DAYFLO TO TRADITIONAL DATABASES

TRADITIONAL DBMS	THE BENEFITS OF DAYFLO
• Fixed record format. All records must look alike.	• Free-Form record format. No two records need look alike. All record formats are stored in same database.
• No word processing capabilities.	• Integral word processing. Create and store letters, memos, notes, ideas, etc.
• Fixed field length.	• Variable field length. No counting character spaces.
• Adding new fields requires remapping or reformatting of database.	• Instantly add new fields to existing records without reformatting the entire database.
• One value per field.	• Multiple values per field. Information where you want it.
• Retrieves data based on pre-planned criteria only.	• Retrieves information based on content or key words.

WHAT YOU CAN DO WITH DAYFLO

- Client Record Tracking
- Sales Lead Tracking
- Personnel Records
- Letters, Memos, Reports
- Form Letters
- Note Taking
- Purchase Order Tracking
- Project Management
- Field Service Tracking
- And much more

Until now.



Automatic Conversion.
OK, you're sold on DayFlo. But what do you do with your present database? Convert it to DayFlo. Use our Automatic Record Conversion program for your pfs or dBASE II data.

Seeing Is Believing.
Visit your nearest DayFlo Dealer and see DayFlo in action for yourself. Or, if you prefer, we'll send you a Demo Disk for $10. For details, call 1-800-7DAYFLO. In California, call 1-800-CDAYFLO.

And please ask questions. Anything your brain, either side, can conjure up.

DAYFLO™

The right brain will love it.

DayFlo, Inc., 2500 Michelson Drive, Building 400, Irvine, CA 92715

CIRCLE NO. 217 ON READER SERVICE CARD

CHECKING OUT THE LIBRARIANS

KURYAN THOMAS

PolyLibrarian is powerful and polished and SR-LIB is unsophisticated and limited, but either librarian is better than none.

One of the most important techniques for writing successful software is modularization—writing small sections of code, each of which performs one specific task, and thoroughly testing these sections before combining them into a single program. Using this procedure, a programmer can develop a collection of basic tools that can be used over and over by different programs. Most compiled languages support this kind of piece-by-piece software development.

Combining these separate modules into one executable program is the task of the linker, a utility provided with all versions of PC-DOS. The modules written by the programmer are combined, and then libraries of functions required by the language being used are searched for other needed modules. Typically, the programmer's modules are turned into "object modules" by the compiler, and the programmer informs the linker which object modules to combine into the final executable file. However, if the linker cannot locate a module needed by the program, it will generate an "UNRESOLVED EXTERNALS" error message, and the linker's output will not be able to be run.

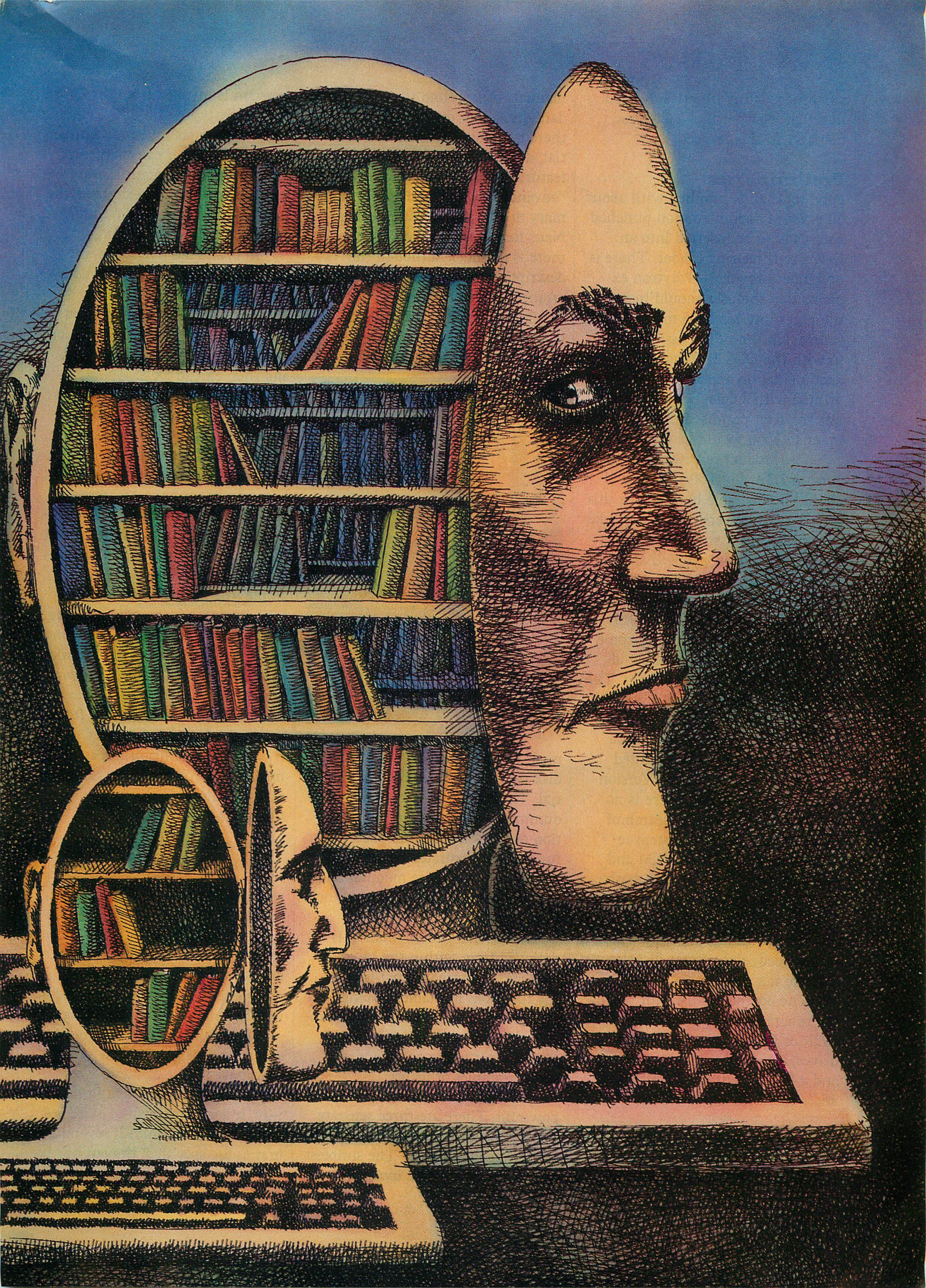
The burden of keeping track of which object modules are required by a particular program is entirely the programmer's. If one object module is inadvertently left out of the linker's list, the linker will not be able to detect an unresolved reference until the very end of the link session. This can be frustrating in the MS/PC-DOS environment, because the linkers are quite slow. It may seem that a solution is to specify every object module that may be needed and hope for the best; the problem with that method is that the linker will combine every module specified into the run file, needed or not, thus increasing the size of the run file.

The ideal solution is to put all related object modules into a single file, which the linker can search for any references it needs—exactly what it does with the language libraries supplied with compilers. For quite a while, utilities that performed this task were not available for MS/PC-DOS. Now, however, there are several such products, appropriately named *librarians*, on the market. In this article, I will look at two of them, PolyLibrarian, from Polytron Corporation, and SR-LIB, from Software Research.

Although the primary purpose of a librarian is to make libraries of object modules, there are other tasks they can and should perform. They must also allow the programmer to replace modules in a particular library with updated versions and to

Kuryan Thomas is a graduate student at Virginia Tech in Blacksburg, VA. His research involves digital image processing.

ILLUSTRATION • MICHAEL GIBBS



examine existing libraries. A good librarian is rather like a database manager in this respect.

Preliminaries

PolyLibrarian is supplied with about 60 pages of documentation punched and ready to be inserted into an IBM-size three-ring binder. There is no binder provided—Polytron explains that the user should insert the pages into his DOS manual. (Of course, it's almost impossible to do so with the DOS 2.10 manual.) The documentation appears to have been printed with a daisy wheel printer, but Polytron promised that the manuals soon would be typeset.

The product reviewed here is version 1.2A, but PolyLibrarian II should be available by the time this article appears (version 1.2A will be discontinued). A subset of PolyLibrarian II, version 1.3, will also be available. All versions require 128K of RAM and 70K of disk space, and all can run under PC-DOS 1.0, 1.1, 2.0, and 2.1. Either a monochrome or a color display can be used, although there is no use of color. The price for versions 1.2 and 1.3 is $99; PolyLibrarian II is $149.

SR-LIB is supplied with nine pages of documentation, printed with a dot-matrix printer. Version 1.03 is reviewed here; there is no mention by Software Research of any planned updates. System requirements are 64K of RAM and one disk drive. I ran it with both DOS 1.1 and 2.1 with no problems. SR-LIB's price is $29.95.

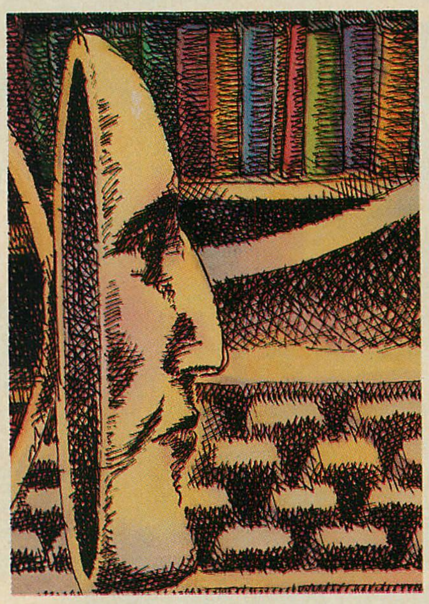
Both programs are supplied on single-sided, eight-sector diskettes that are not copy protected.

The computers used for testing the programs were an IBM PC with 192K of RAM, two single-sided drives, and a color graphics adapter and a PC/XT with 192K of RAM and a monochrome monitor.

PolyLibrarian

PolyLibrarian's documentation is outstanding in a sea of mediocre manuals. Apart from the now-standard disclaimer of any liability, there is a table of contents, a tutorial, detailed descriptions of all the features available, coverage of some "advanced topics," and a brief summary of all commands in Backus Naur format (BNF). Regrettably, there is no index, although the descriptions of the features are arranged alphabetically.

The tutorial provides a good introduction to the power of this program. There are several "object



modules" provided on the program disk that are to be used for the tutorial. These modules are merely dummies—they perform no function outside the tutorial. Each major function is illustrated by an example that uses these modules and that the user can key in and try out if he desires to do so.

Additional documentation is provided from within the program by a HELP command that lists all the functions available. Typing HELP <command name> will provide the user with a short description of a particular command and some examples of its use.

The program itself is extremely powerful. There are three separate ways to issue commands to it, all described in the manual. The simplest way is the *interactive mode*. The program is invoked from the DOS prompt by typing in "LIB"; when it loads, it prompts for input with an asterisk. The user can then enter commands in a sequential way. After each function is performed, the asterisk returns. A QUIT command exits to the DOS prompt.

In *command-line mode*, the program invocation is followed by whatever commands the user wishes to be executed, separated by "-" symbols. The program will load, execute each command in order, and then exit to DOS.

The third mode is *command-file mode*. Here, the invocation command is followed by the name of a file that contains commands to be executed. These commands will be executed, and the program will return to the DOS prompt. In addition, all three modes can be mixed (this is one of the "advanced topics" discussed in the manual). For example, the program can execute some commands in command-line mode, then some from a command file, then go into interactive mode to accept commands directly from the keyboard, and, finally, it can execute some more commands from the original command line.

Table 1 is a list of the commands that PolyLibrarian recognizes. Qualifiers exist for some functions—for example,

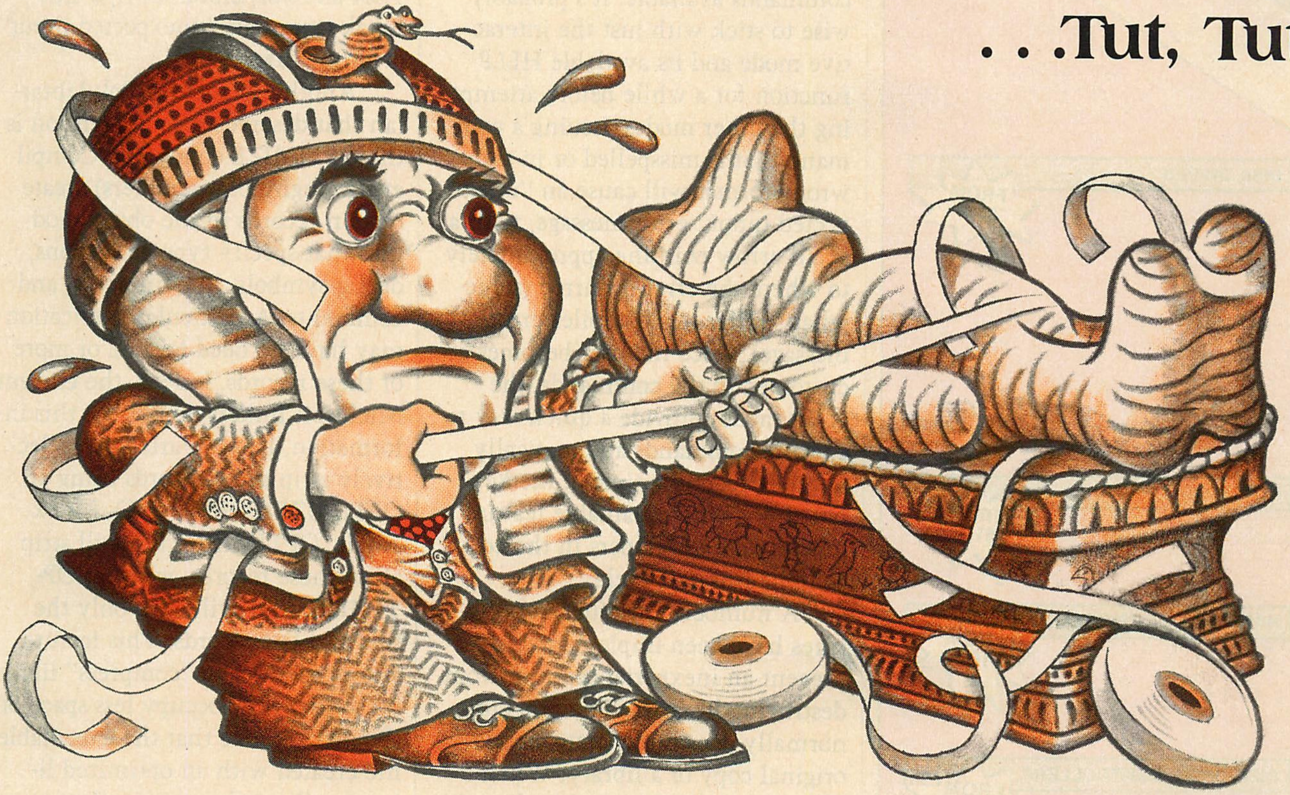
ADD name TO library NOASK RETAIN

will add name.obj to library.lib, making it the default library (that is, automatically SELECTing it), and will retain the module name given by the compiler. If a module already exists with the same name, it will be deleted without ASKing.

These qualifiers can be in any order; there is a command parser that makes sense of them. Every time a library is specified in this way, it is automatically SELECTed as the current library. If several libraries are specified, the last one becomes the current library on exit.

Considering Tape For Backup Storage?

...Tut, Tut.



There was a time when tape was the way to store backup data from your hard disk. It was fine for archives, but not when you wanted your data live and on-line. Today's business pharaoh can have his data on-line and backups, too—with 5 or 10 megabyte **removable cartridge** hard disk systems from **Micro-Design**.

Single-function tape backup units can't come close to matching the versatility and cost-effectiveness of hard disk cartridges; you get backup capabilities PLUS the benefit of having another full 5 or 10 megabytes of on-line data access.

Use hard disk cartridges the same way you use a floppy disk system. Just switch them out as needed—one cartridge for wordprocessing, another for accounting, and another for backup storage. Add additional cartridges whenever you like.

Fast. Cartridges run as fast, or faster, than fixed hard disk drives. A 5 megabyte backup takes only a few minutes, using Micro-Design's backup utility software. Switching out cartridges takes just a few seconds.

Convenient and Secure. It's easy to lock-up sensitive files—just remove the cartridge and put it somewhere safe. If you share a computer with others, everyone can have their own cartridge. Or, have a home cartridge and a business cartridge for use with your portable computer.

IBM® International Business Machines.

Unlimited Expansion. No more erasing old data to increase space on your hard disk. Fact is, there's no limit to the amount of off-line storage you can add.

Our **DATA Series** 5 and 10 megabyte cartridge hard disks are external drives available in removable only, or paired with 10, 20, or 33 megabyte fixed drives. Our **SQ Series** features 5 megabyte cartridge drives, in matching external cases or ready for internal installation. All are 100% compatible with the IBM-PC, XT, and work-alikes, with no software modifications needed. You can even boot from the hard disk.

Don't let single-function hard disk and tape units send your asp up the Nile. Ask your local computer dealer about Micro-Design's versatile hard disk cartridge units. Or, call us for details on our full product line and ordering information. Dealer inquiries invited.

We've got a complete line of mass storage devices, including floppy drives. Prices start as low as **$1549** for a 20 megabyte fixed hard disk drive. Call for our **FREE** catalog today!

One year warranty. Network compatible. Backup utility software included with 2 drive systems.

MICRO-DESIGN MASS STORAGE DEVICES

Micro-Design

Making The Most of Your Micro.

CIRCLE NO. 134 ON READER SERVICE CARD

Call Toll FREE 1-800-531-5002.
In Texas, call 512-441-7890.
6301 B Manchaca Rd., Austin, TX 78745

All commands (except KILL) can be specified by typing in their first one or two characters.

There is quite a repertoire of commands available. It's probably wise to stick with just the interactive mode and its available HELP function for a while before attempting the other modes. Issuing a command that is misspelled or in the wrong format will cause an "unexpected input" error message, with a small arrow pointing approximately to where the error occurred (somewhat similar to a compiler error message). Once the user becomes accustomed to the commands, the other modes provide a quick way to execute commands automatically without spending time at the keyboard, which is a valuable feature for software developers in the edit-compile-link-test cycle.

A number of useful safety features have been implemented to prevent an inexperienced user from destroying libraries. The program normally does not operate on the original copy of a library. When a library is SELECTed (either directly by use of the SELECT command or indirectly by some other command), a back-up copy is made on the default disk (the WORK command allows another drive to be specified for this purpose). All operations are performed on this copy. Thus, if something goes wrong, the original library is still available. If the user inadvertently issues a command he didn't intend to, he can recover by typing KILL. This wipes out the working copy, leaving the original library. The UPDATE and QUIT commands make the (altered) back-up copy into the actual library. The original library is not saved. This back-up feature can be disabled by issuing a BACKUP NO command. Polytron strongly discourages this, as I do, because without the back-up, any changes made to the library will be permanent.

One potential problem with version 1.2A is that it will respond to a Ctrl-Break by exiting immediately to DOS, without updating the current library. Even that problem has been taken care of with version 1.3A and PolyLibrarian II; a Ctrl-Break causes an "unexpected input" error message.

Another feature of PolyLibrarian that deserves special mention is the OPTIMIZE command. Compilers (though not assemblers) create certain records in the object modules that specify type definitions, debug symbols, line numbers, and comments. A particular application may have no need for one or more of these records. Indeed, the current Microsoft linkers do not use this information. PolyLibrarian can selectively strip these records using an OPTIMIZE COMMAND. For example, OPTIMIZE 1111 will strip all of these records, whereas OPTIMIZE 1001 will strip only the first and last records. This feature allows the user to "compress" libraries so that they occupy less space on the disk. Notice that the executable file created with an optimized library will not be any smaller.

Programmers who own one of the new Microsoft compilers (Pascal 3.2 and FORTRAN 3.2) know that the libraries provided with them are radically different from previous versions. The supplied linker must be used, because the PC-DOS linker cannot read these libraries. This accounts for the major difference between PolyLibrarian versions 1.2A and 1.3. Version 1.2A could not read the new libraries, which prevented curious users from examining them. Version 1.3 corrects this by adding a MODE command that allows the user to specify, among other things, whether the library is to be Microsoft 3.2- or 3.1x-compatible. PolyLibrarian II goes one step further and supports the Intel library format, even allowing Intel modules to be added to Microsoft libraries and vice versa. I was not able to test this feature because I don't own an Intel compiler.

Capable COBOL.

For over twenty years, COBOL has proved it can handle the most awesome data handling requirements of the business world.

And Microsoft® COBOL brings this power to the MS™-DOS environment in an even more flexible version. It's the ideal choice for transporting and adapting the thousands of mainframe and minicomputer programs now on the market.

Microsoft COBOL was designed after the ANSI COBOL standard. Its quality and performance has been fully certified by the GSA.

Microsoft COBOL includes a complete set of file handling facilities including support for sequential, line sequential, relative and B+ tree indexed sequential (ISAM) files.

MICROSOFT®
The High Performance Software

The advanced screen handling features, compatible with Data General®, make it easy to create forms and menus for interactive user programs. Direct cursor positioning, highlighting, automatic format conversion, projected fields, and automatic field skip are just some of the capabilities.

An interactive source code debugging tool allows you to step through your program in symbolic form. There is no need to wade through object code and addresses in memory to see what is going on. It's a complete development package.

Applications developed with the compiler can be distributed without a runtime support charge.

Like all Microsoft languages, the standard linking interface makes it easy to combine assembly language subroutines.

Call 800-426-9400 to order the capable COBOL. $700.*

In Washington State, call 206-828-8088. Ask for operator I6, who will rush you your order, send you more information, or give you the name of your nearest dealer to see Microsoft COBOL in action.

*Price exclusive of handling and Washington State sales tax.
Microsoft is a registered trademark and MS is a trademark of Microsoft Corporation.
Data General is a registered trademark of Data General Corporation.

Version 1.3's MODE command also allows date- and time-stamping of each entry in a library for archival purposes. These stamps are derived from the DOS clock.

Version 1.3 and PolyLibrarian II support full DOS 2.xx path names for every function that requires them. They all seemed to function smoothly. Version 1.2A has no DOS 2.xx support. My pre-release version of PolyLibrarian II can be run only under DOS 2.xx, but Polytron claims that its shipped version can be run under earlier versions of DOS as well (without any path-name support, of course).

SR-LIB

This small librarian is considerably more limited than PolyLibrarian. Only two operations are supported: adding and deleting modules from a library. Nine pages of documentation may seem rather few, and those pages are unattractively put together, but they do a good job of explaining how to use the program. There is no tutorial, but keep in mind that with only the two basic functions of a librarian supported, there is little need for one.

An example showing how to specify commands to the program is provided in the SR-LIB manual. It also includes a list of error messages, which are two-letter acronyms—for example, "im" signifies "invalid module." The brief introduction explains the benefits of using libraries and provides some interesting information—for example, it states that PC-DOS libraries have enhanced indices over Intel libraries.

The operation of SR-LIB is similar to, though not quite as flexible as, PolyLibrarian's interactive mode. When the program is invoked from DOS by typing in SRLIB, three successive prompts for input are displayed. The first asks for a library name to work with. The second asks for a list of commands, and the third asks whether a listing of the specified library is required. If a library that does not exist is specified on the first prompt, the program will ask whether a new library should be created. Pressing "y" will create a new library; pressing anything else will halt the program. After the commands entered at the second prompt are executed, a listing will be generated on the screen (if CON was typed in response to the second prompt) or on the printer (if PRN was typed). The program then exits to DOS. Unlike PolyLibrarian, SR-LIB cannot work on more than one library in one session. The program for each library must be invoked.

The operations are entered in response to the second prompt as a sequence of object module names separated by a plus sign or a minus sign. A plus sign means "add the following object module to the library," and a minus sign signifies deletion. There is no way to explicitly replace a module—it must be deleted and then replaced.

Any of the prompts can be answered with a null entry simply by pressing ENTER without any input (even the library name prompt). To get a listing of a library without doing anything to it, the user can give a null response to the operations prompt and indicate the desired output device at the listing prompt. (A null response to the listing prompt will prevent a listing from being produced.) The listing does not include any information on public and external symbols, and it cannot be sent to a disk file.

No optimization can be specified. I examined some libraries created by SR-LIB using PolyLibrarian and discovered that the optimization levels were all 0000, meaning that no optimization was done.

Neither DOS 2.xx path names nor the new Microsoft library format is supported by SR-LIB.

Complaints and Bugs

I have a small complaint about the explanation of command-line mode

Potent Pascal.

Microsoft® Pascal may be the most powerful software development environment available for the MS™-DOS system. It combines the programming advantages of a structured high-level language with the fast execution speed of native code compilation.

And it exceeds the proposed ISO and ANSI standards with logical extensions that make the language more powerful and versatile. For example, programming capabilities even allow you to manipulate data at the system and machine level.

It gives you single and double precision IEEE floating point arithmetic. Numeric operations take advantage of the 8087. Or automatic software emulation is provided if the coprocessor is not installed.

MICROSOFT.
The High Performance Software®

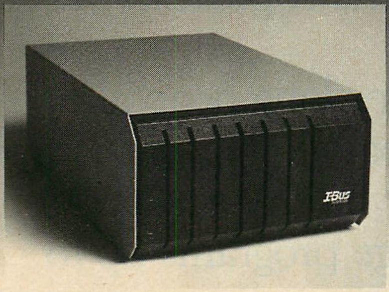
Support for long heap allocation and separate module compilation gives you the flexibility to create large programs up to one megabyte.

And the standard linking interface makes it easy to combine Microsoft FORTRAN or assembly language subroutines.

Call 800-426-9400 to order the potent Pascal. $300.*

In Washington State, call 206-828-8088. Ask for operator C5, who will rush you your order, send you more information, or give you the name of your nearest dealer to see Microsoft Pascal in action.

*Price exclusive of handling and Washington State sales tax.
Microsoft is a registered trademark and MS is a trademark of Microsoft Corporation.



COMMAND	FUNCTION	COMMENTS
LIB	Invokes the program from DOS	Can be followed by additional commands
ADD	Adds a module	Can be used with additional qualifiers
BACKUP	Displays or changes current back-up status	Turning it off is *not* recommended
CASE	Specifies whether names should be case-sensitive	
CHANGE	Changes a module or public name	
DATE	Specifies whether modules should be date-stamped	Archival information
DELETE	Removes a module	
DIR	Directory	Same as DOS DIR/W. Supports wildcards
EXTRACT	Makes an object module from a library module	Can be used with NDASK
FIND	Finds a public name or external	Only 1.3 or II
HELP	Provides help	Can be followed by a specific command name for help on that function
INTERACTIVE	Enters interactive mode	Only useful in command line or command file modes
KILL	Discards changes since last UPDATE	
LIST	Lists all modules	Optionally, publics and externals. Cannot be sent directly to printer, but can be sent to a file.
MAKE	Creates a new library	Can be used with NOASK
MODE	Sets defaults for TYPE, RETAIN, DATE, and OPTIMIZE	Only in 1.3 and II
OPTIMIZES	Compresses libraries	Selective compression is possible
QUIT	Exits current mode	Automatically UPDATEs
REPLACE	Replaces a module with another	
SELECT	Selects the working library	
TYPE	Specifies which kind of library (old or new Microsoft format)	Only 1.3 and II
UPDATE	Makes changes permanent	
WORK	Specifies which drive to use for back-up copy	

Table 1: PolyLibrarian Functions

in PolyLibrarian. The manual for version 1.2A explains that commands should be separated by slashes or minus signs. However, it is not explicitly stated that a space must precede these separators. Polytron intends to clarify this in its new documentation.

SR-LIB could not read the new Microsoft libraries, but instead of giving an "im" (invalid module) error and stopping, it gave the message and froze the system, requiring a reboot. I should note here that SR-LIB *will* make libraries that the new Microsoft linker can read—the language libraries that are supplied with the compilers are what the linker is unable to read.

Some error messages produced by SR-LIB are not described in the manual. For example, giving a null response to the library name prompt will cause the program to ask if the user wants to create a new library. If the user replies "yes," the program reports "creation error." These errors are self-explanatory, but they should still be listed with the two-letter messages in the manual.

Ferocious FORTRAN.

Microsoft® FORTRAN crunches numbers with a vengeance!

It combines fast and efficient native code compilation with built-in 8087 coprocessor support. The result? Mini and mainframe performance from your MS™ DOS micro.

Based on the '77 standard, Microsoft FORTRAN supports extensive statements and data types—including complex numbers and IEEE single and double-precision floating point accuracy.

Support for large arrays (greater than 64K bytes), separate module compilation, and overlays, allow you to create very large programs—up to one megabyte, with access to more than 65 thousand records in a file as large as four gigabytes.

MICROSOFT.
The High Performance Software

How do programmers feel about Microsoft FORTRAN?

"The first FORTRAN compiler that takes advantage of the full addressing capability of the 8088 and the power of the 8087."
—Jack Wilschke, *Softalk*

"We decided to use the Microsoft FORTRAN Compiler for its INTEGER 4 capability and the flexibility of its 8087 implementation."
—Charlie Huizena & Chip Barnaky, *PC World*

Call 800-426-9400 to order the ferocious FORTRAN. $350.*

In Washington State, call 206-828-8088. Ask for operator C4, who will rush you your order, send you more information, or give you the name of your nearest dealer to see Microsoft FORTRAN in action.

*Price exclusive of handling and Washington State sales tax.
Microsoft is a registered trademark and MS is a trademark of Microsoft Corporation.

Comparing the Librarians

Comparing these two markedly different programs is a bit like (if you'll pardon the puns) comparing the Library of Congress to the public library in my small town. The Library of Congress contains many millions of books, making it hard to find any particular book; my public library has only a few thousand books, so it is possible to look over all of them in one weekend.

In much the same way, PolyLibrarian is an extremely powerful software development tool, but it may take the user 45 minutes with the tutorial and three weeks of use before he is familiar with all the features. In contrast, it takes about 15 minutes to learn to use SR-LIB to its full (but limited) capacity. SR-LIB performs the basic functions required of a librarian, provided it doesn't have to work with the new Microsoft compilers' libraries.

It is no small feat for a microcomputer program that costs $99 ($149 for II) to hold its own against a supermini program that costs many thousands or tens of thousands of dollars. PolyLibrarian is one software product that can't be outgrown.

People who often work with large numbers of object modules need PolyLibrarian. It can contain approximately 8,000 modules in one library and an unlimited number of public and external symbols in each module (this information is from the manual—I haven't the stomach to try it out!).

SR-LIB has a limit of 500 modules and 1,024 public symbols. The limitation on the number of modules isn't serious, but more than 1,024 public symbols can be created easily in a large library of scientific subroutines (in FORTRAN, for example, each COMMON block name is a public variable).

Which One Is Better?

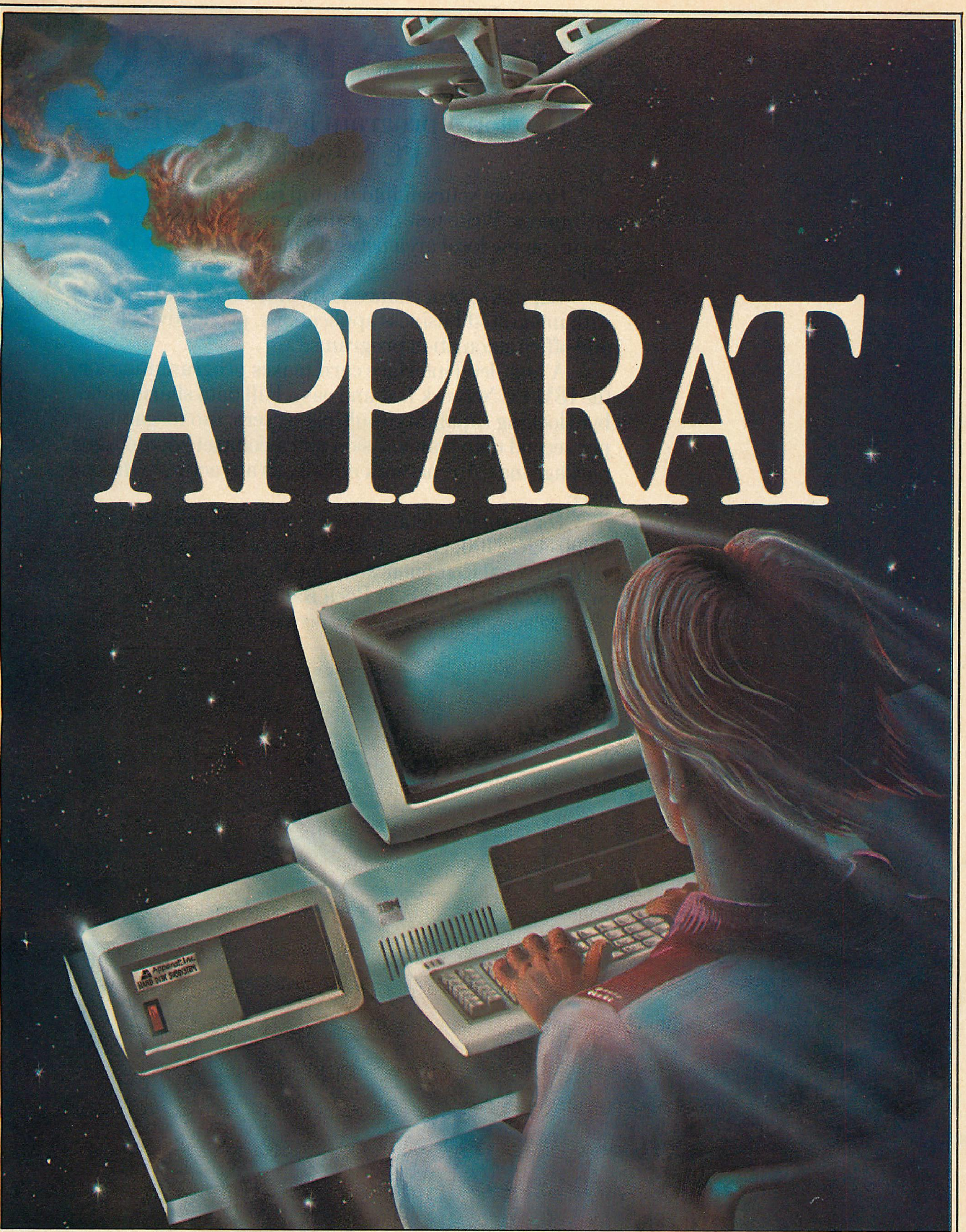
For serious programming, a librarian should be considered as important as a compiler. Even an unsophisticated librarian is better than none. I can thus recommend SR-LIB, especially considering its low price, for programmers whose programming needs are not large and who have a limited budget for software tools. SR-LIB does the basic job well.

PolyLibrarian is a powerful and polished product. Before becoming involved with micros, I worked with superminicomputers, such as the DEC VAX 11/780. These machines are usually supplied with powerful software development tools. However, PolyLibrarian is superior in some respects to the VMS librarian supplied with the VAX. Not only is PolyLibrarian easier to use but it also has features missing in the VMS Librarian, such as extensive archival capabilities.

It is no small feat for a microcomputer program that costs $99 ($149 for II) to hold its own against a supermini program that costs many thousands or tens of thousands of dollars. Polytron deserves credit for this outstanding product. PolyLibrarian is one software product that can't be outgrown.

PolyLibrarian
Polytron Corporation
P.O. Box 787
Hillsboro, OR 97123
503-648-8595
Version 1.2A—$99.00
PolyLibrarian II—$149.00
CIRCLE 496 ON READER SERVICE CARD

SR-LIB
Software Research
P.O. Box 10004
Austin, TX 78766
$29.95
CIRCLE 497 ON READER SERVICE CARD



APPARAT TAKES YOUR PC TO NEW REALMS.

4401 South Tamarac Parkway, Denver, CO 80237—Customer Service 303/741-1778
Apparat retail outlets—DENVER: 4401 S. Tamarac Pkwy. 303/771-2032 • 5224 W. Warren Ave. 303/985-1217
CHICAGO: 1844 S. Arlington Heights Rd. 312/640-0322

CIRCLE NO. 106 ON READER SERVICE CARD

Join the network

Develop a program for the new IBM PC Network.

Promote yourself into the top ranks of network programmers. Write new programs or modify existing ones to run on the local area network that's sure to be a best seller.

Capitalize on a chance to be in from the beginning with integrated business applications, productivity tools and office automation programs.

A high-level interface card makes programming for the IBM PC Network fast and easy. Full data sharing and byte locking capability, multiple servers, two megabits per second data transmission and a complete diagnostic package make the network equally as attractive to the end user.

For complete details about joining the network call 1-800-426-2700. Or write IBM Corporation, Editor, IBM Personal Computer Seminar Proceedings, 5Q9/4629, P.O. Box 1328 Boca Raton, FL 33432.

CIRCLE NO. 135 ON READER SERVICE CARD

Little Tramp character licensed by Bubbles Inc., s.a.

IBM PC Network Specifications

IBM PC NETWORK ADAPTER

HARDWARE HIGHLIGHTS

MICROPROCESSORS
- 6MHz 80188
- 6MHz 82586

MEMORY
- 32K PROTOCOL ROM
- 16K RAM
- 8K NET BIOS ROM

VIDEO COMPATIBLE RF MODEM
- TRANSMIT 50.75 MHz (CH T14)
- RECEIVE 219 MHz (CH J)
- SUPPORTS 1000 NODES
- MAXIMUM DISTANCE 5KM RADIUS FROM HEADEND
- MULTIPLE SERVICES POSSIBLE

DIAGNOSTICS
- POWER-ON SELF-TEST
- ON-LINE MEDIA MONITORING

OTHER
- SUPPORTS DMA DATA TRANSFERS
- 2-MEGABIT/SECOND DATA RATE
- MID-SPLIT BROADBAND

FIRMWARE HIGHLIGHTS

OPEN ARCHITECTURE
- PEER-TO-PEER NETWORK
- OPERATING SYSTEM INDEPENDENT
- LOCALNET/PC™, PUBLISHED LAYERED PROTOCOL

FUNCTIONS
- BASE FUNCTIONS PROCESSED ON THE ADAPTER, NOT THE PC
- DISTRIBUTED NAME SUPPORT
- REMOTE PROGRAM LOAD
- 32 CONCURRENT TWO-WAY SESSIONS
- HIGH THROUGHPUT RATE AT SESSION LAYER
- CHARACTER SET INDEPENDENT

IBM PC NETWORK CABLING SYSTEM

IBM PC NETWORK TRANSLATOR UNIT

HARDWARE HIGHLIGHTS
- SINGLE RF CHANNEL CONVERSION
- ATTACHMENT OF UP TO 72 PCs WITH IBM CABLING
- ATTACHMENT OF UP TO 256 PCs WITH CUSTOM CABLING
- DATA ONLY
- ALLOWS NODES WITHIN A 1000-FOOT RADIUS

CABLE

TYPE
- STANDARD CATV MEDIA (75 OHM COAX)
- TREE TOPOLOGY
- CATV STANDARD F-CONNECTORS
- PREBALANCED BROADBAND NETWORK

KITS
- BASE EXPANDER (ALLOWS EXPANSION FROM TRANSLATOR)
- SHORT-DISTANCE KIT (1 FOOT ADDITIONAL CABLE)
- MEDIUM-DISTANCE KIT (400 FEET ADDITIONAL CABLE REQUIRED)
- LONG-DISTANCE KIT (800 FEET ADDITIONAL CABLE REQUIRED)
- CABLE AVAILABLE IN 4 LENGTHS: 25 FT., 50 FT., 100 FT., AND 200 FT.

MAXIMUM PCs AND DISTANCES SUPPORT	RADIUS FROM TRANSLATOR	PCS
• TRANSLATOR ONLY	200 FEET	8
• 8 SHORT-DISTANCE KITS	200 FEET	72
• 8 MEDIUM-DISTANCE KITS	600 FEET	72
• 8 LONG-DISTANCE KITS	1,000 FEET	72
• 8-KIT COMBINATION	200 to 1,000 FEET	72

IBM PC NETWORK SOFTWARE

DOS 3.1

EXPANDED SUPPORT FOR NETWORKING
- FILE SHARING
- RECORD LOCKING DOWN TO BYTE LOCKING

PROGRAM INTERFACE TO NETWORK SOFTWARE
- REDIRECTION CONTROL
- INSTALLATION CHECKING
- DIRECT EXECUTION OF NET BIOS FUNCTIONS
- MULTIPLE SERVERS

IBM PC NETWORK PROGRAM

FULL SCREEN INTERFACE AVAILABLE

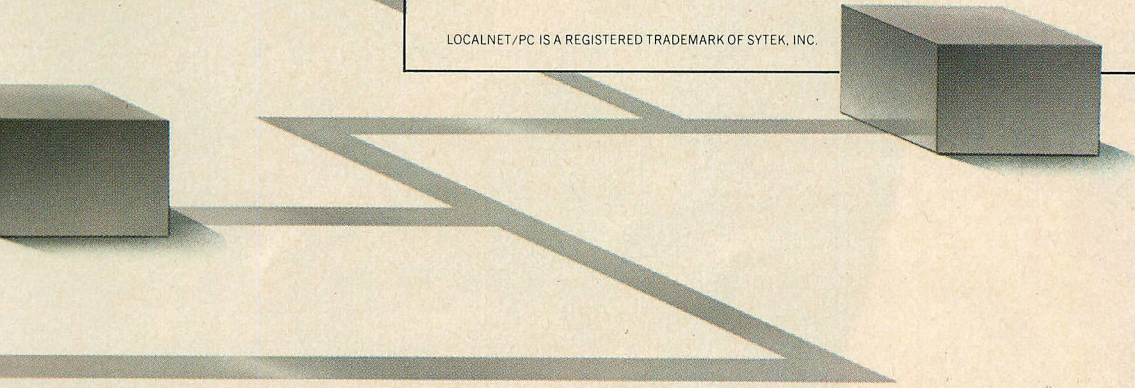
REDIRECTOR
- ALLOWS USE OF SHARED PRINTERS
- ALLOWS USE OF SHARED DISKS AND DIRECTORIES
- PROVIDES CAPABILITY TO SEND MESSAGES

FILE SERVERS
- SHARED USE OF NAMED DISKS OR SUBDIRECTORIES
- PASSWORD PROTECTION AGAINST UNAUTHORIZED ACCESS
- VARIETY OF ACCESS MODES SUPPORTED (E.G. READ ONLY)
- RECORD LOCKING TO CONTROL MULTIPLE UPDATES

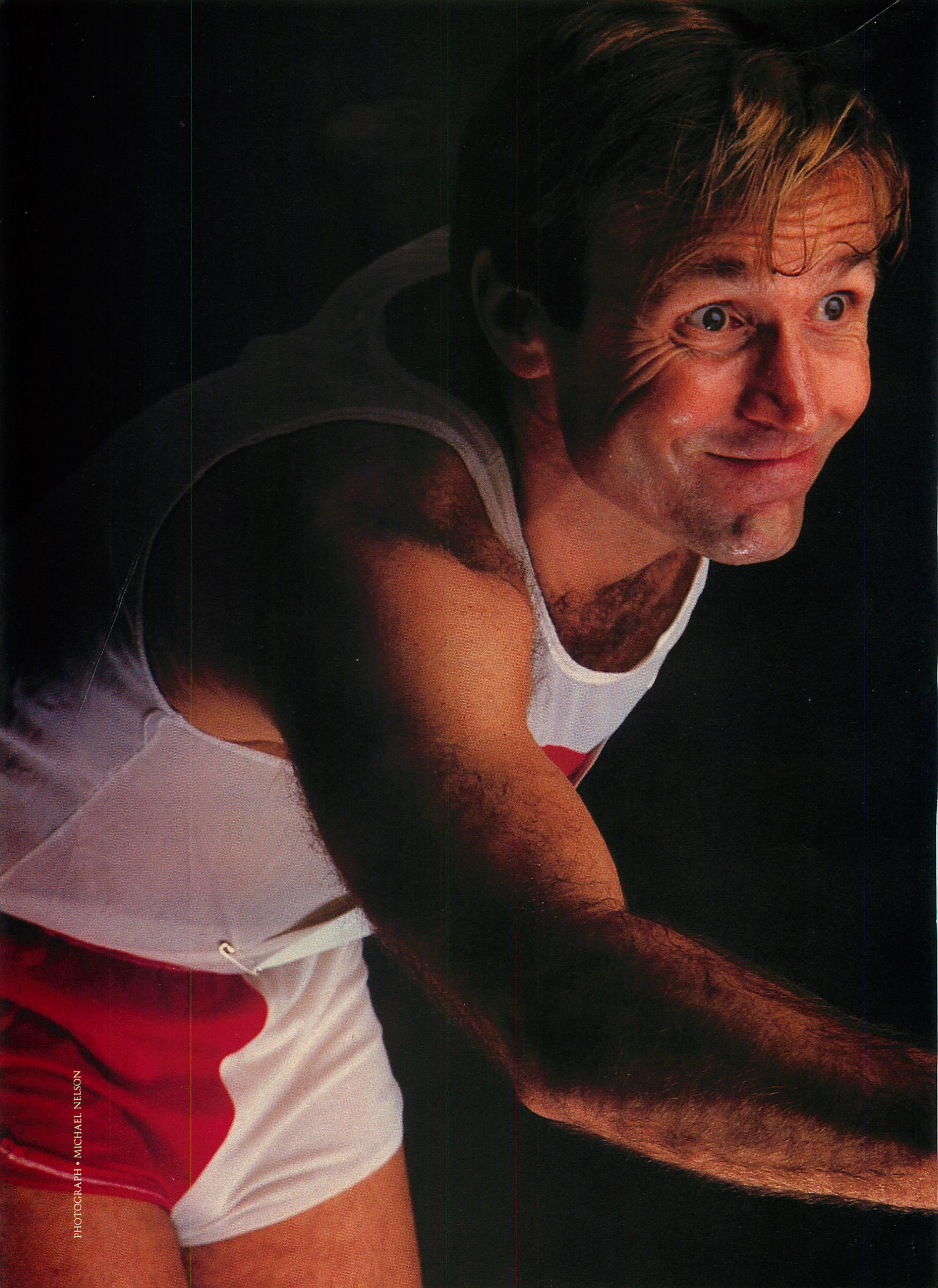
PRINT SERVER
- SHARED USE OF NAMED PRINTERS
- PASSWORD PROTECTION AGAINST UNAUTHORIZED ACCESS
- AUTOMATIC SPOOLING AND QUEUING OF OUTPUT
- QUEUE MANAGEMENT FACILITIES ON SERVER STATION

MESSAGE SERVER
- INTERACTIVE MESSAGE EDITING/TRANSMISSION/RECEPTION
- PRESERVES FOREGROUND APPLICATION CONTEXT
- AUTOMATIC NOTIFICATION OR LOGGING TO DISK/PRINTER

LOCALNET/PC IS A REGISTERED TRADEMARK OF SYTEK, INC.



IBM®



PHOTOGRAPH • MICHAEL NELSON

INSTANT RELAY

Beat the clock with this feature-packed communications package.

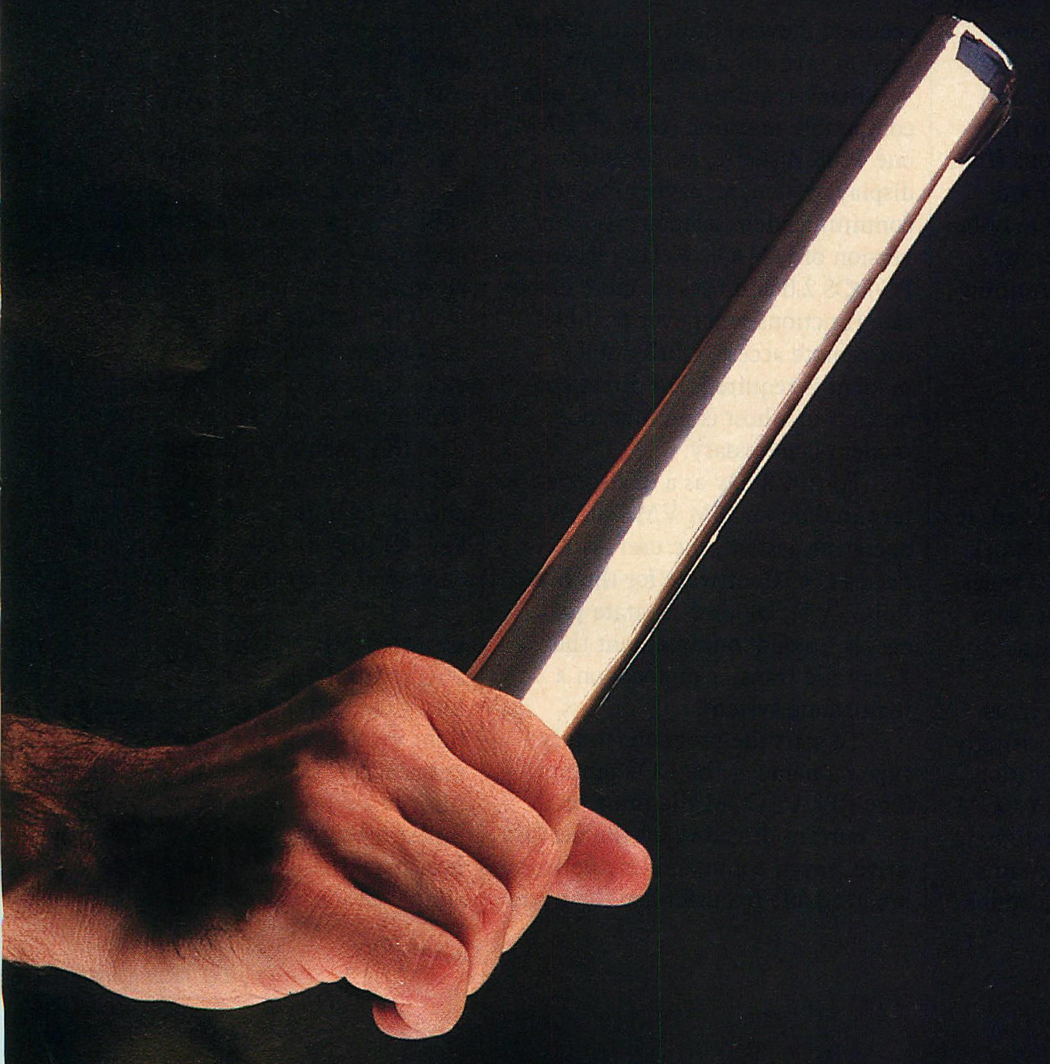
AUGIE HANSEN

The deadline for a new product delivery approaches, and programmers on both coasts are hard at work on some last-minute additions. In the wee hours of the morning an interesting exchange takes place between two programmers, each typing feverishly in an attempt to beat the clock, as they have done so many times before.

"I put the mods you asked for into the disk fixer program. Do want to look 'em over?"

"Yeah—just send me the three modules for the sector modification mode. The rest of it looked OK to me last time. I'm printing out the updated spec sheets as we speak."

Augie Hansen is a programmer for a major telecommunications company and a contributing editor to this magazine.



"OK. They'll be on their way as soon as the file you're sending me is done. After I start the transfer I'm going to do some editing on the user guide. I'll check back when the transmission's complete. Beep if you need anything else."

A PC-to-PC "conversation," error-free file transfers, printing, and editing all going on simultaneously. At least it seems that way. Is this possible? What miracle of modern technology is at work here?

It's Relay, a powerful communications program for the IBM PC by VM Personal Computing. In addition to the features already alluded to above, Relay features terminal emulation (VT100), several varieties of file transfer protocols (including XMODEM), keyboard customization, an unattended answer mode, a dialing directory, script commands, translation tables, a full screen editor, and extensive context-sensitive help. A big plus for IBM mainframe users is that a 3270 mode is provided—this mode, when used with PC3270 (another VMPC program) on the host machine, permits IBM 3270-series terminal operations from the PC without any special hardware. Also, files of any type may be transferred between a PC running Relay and a host computer running Relay/VM (formerly called PLEASE). The version that was tested for this review is 2.3.

GETTING STARTED

Relay is not copy protected, but each master diskette is serialized. It may be copied easily onto a "working" diskette or a hard disk. Copies of the program that have the same serial number are not permitted to talk to each other.

The program is written in assembly language and is surprisingly small for what it does. In addition to the main program, RELAY.COM (about 44K), there is a separate help file and a set of files that contains keyboard macros, communications scripts for a wide range of services, and miscellaneous utility programs.

Two conversion programs are provided for users whose hosts don't have the capability of doing binary file transfers. These programs do binary-to-hex and hex-to-binary conversions for use with simple ASCII file transfer functions. Because the transfer techniques provided in Relay are faster and more accurate, however, users are encouraged to take advantage of them.

Relay requires that a name (the PC ID) be given to permit calling and answering systems to identify each other. The PC ID is input on a New User screen the first time Relay is used, and then a Modem screen asks for the type of modem.

Relay requires an IBM PC or a compatible machine, 128K of RAM, one 160K diskette, an 80-column display system, an asynchronous communications adapter, and any version of PC-DOS. Relay recognizes PC-DOS 2.0 and uses its more elaborate functions whenever possible. For remote access, a full-duplex modem is required. For direct connection to a host computer, no modem is necessary.

To run Relay as a 3270-series terminal on an IBM VM/370 SP operating system, the user will also need PC3270 software for the host. Relay/VM, another separate package, is needed to implement the Relay file transfer protocol on a mainframe system.

To start the program, simply type its name at the DOS prompt. The RELAY.COM file must be in the current directory on the default drive. Seven command-line options are available to customize the operation of Relay for unattended operation, automatic "load and go" operation, and various display system configurations. The menu presented upon start-up depends on the options specified and on whether the PC ID entry is filled or empty.

Relay requires that a name (the PC ID) be given to permit calling and answering systems to identify each other. The PC ID is input on a New User screen the first time Relay is used, and then a Modem screen asks for the type of modem that is attached. For all subsequent start-ups, unless told otherwise, Relay goes immediately to the OFFLINE menu (see photo 1), from which the user may select one of the primary functions.

Function key <F3> selects the function "Review the Directory of Computers," which is, in essence, a dialing directory (see photo 2). The user can point the cursor at an entry and call it by pressing <F1> or wait for a call from the named entry by pressing <F2>. Function key <F3> is used to create a new entry that is initially a copy of the currently selected entry. This copy may then be edited and named using simple commands. Finally, <F4> causes a summary of connect options to be displayed for viewing or editing.

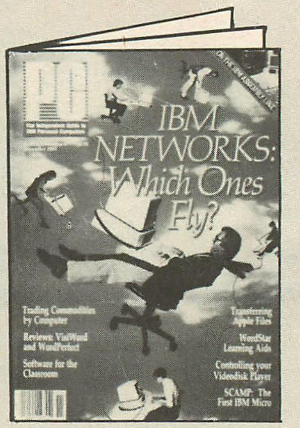
The <Esc> key is used to back out of any screen, restoring the previous context. The only place in which this is not the case is the ONLINE menu, which requires an <F1> selection to return to the current terminal screen. Function key <F10> always gets help, as is explained in greater detail later.

COMMUNICATIONS PROTOCOLS

Relay uses a variety of communications protocols depending on the circumstances. Garden-variety communications with information utilities, bulletin boards, and miscellaneous host machines are handled by a simple character protocol and may

PC's $2995 COMPAQ SWEEPSTAKES

Win a COMPAQ Portable Microcomputer system by entering PC™'s SWEEPSTAKES!



You have the opportunity to win a fully portable IBM compatible system weighing less than 28 lbs...giving you:

- Full function keyboard!
- Systems box housing the CPU Chassis, 9″ phosphor screen and one 320K 5.25″ double-sided disk drive!
- An MS-DOS 1.1 IBM PC Compatible system...featuring 16 bit 8088 CPU, 128K RAM expandable to 512K, Microsoft Basic included. RGB output along with NTSC. Parallel port and 3 expansion ports!
- The COMPAQ Portable will run on over 95% of the software written for the IBM PC!

ENTER TODAY! NO PURCHASE NECESSARY!

BE A WINNER ALL YEAR LONG...SUBSCRIBE TO *PC*, The Independent Guide to IBM Personal Computers today and pay *less than $1.10* a copy!

ENTER *PC*'s SWEEPSTAKES AND YOU MAY WIN A $2,995 IBM COMPATIBLE PORTABLE MICROCOMPUTER!

NO POSTAGE NECESSARY IF MAILED IN THE UNITED STATES

BUSINESS REPLY MAIL

FIRST CLASS PERMIT NO. 66 BOULDER, CO

POSTAGE WILL BE PAID BY ADDRESSEE

PC™'s SWEEPSTAKES
P.O. Box 2442
Boulder, Colorado 80321

OFFICIAL RULES:

No Purchase Necessary

1. On an Official Entry Form or a 3″ x 5″ piece of paper, hand-print your name, address and zip-code. Enter as often as you wish, but mail each entry separately to *PC*'s SWEEPSTAKES, P.O. Box 2442, Boulder, Colorado 80322. Entries must be received no later than January 2, 1985 and the drawing will be held January 31, 1985. All entries become the property of *PC* Magazine, which reserves the right to reprint the name and address of the winner.

2. The winner will be selected in a random drawing from among all entries received, under the supervision of the publishers of *PC*, whose decision will be final. Only one prize will be awarded in this Sweepstakes. Winner will be notified by mail and may be required to execute affidavit of eligibility and release. Odds of winning will depend on the number of entries received. Ziff-Davis will arrange delivery of prize. Taxes are the responsibility of the winner. Any manufacturer's claims and warranties will apply, but Ziff-Davis makes no claims or warranties with regard to any prizes. Prize is not transferable. No substitutions for prizes.

3. Sweepstakes open to all residents of the U.S., its territories and possessions, except employees (and their families) of Ziff-Davis Publishing Company, its affiliates, and its advertising and promotion agencies. Void wherever prohibited or restricted by law.

4. For the winner's name, send a stamped, self-addressed envelope to *PC*'s SWEEPSTAKES, Circulation Department, One Park Avenue, New York, NY 10016.

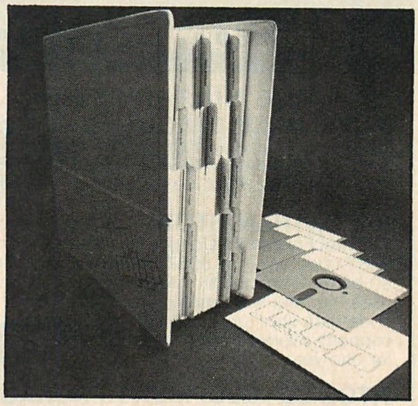
For your IBM/PC

mbp COBOL: 4 times faster, and now with SORT & CHAIN.

$750.

mbp COBOL can be summed up in one word: fast.

Because it generates native machine language object code, the mbp COBOL Compiler executes IBM/PC* programs *at least* 4 times faster (see chart). Fast also describes our **new SORT**, which can sort four-thousand 128-byte records in less than 30 seconds. A callable subroutine or stand-alone, 9 SORT control fields can be specified. And our **new CHAIN** is both fast and secure, conveniently transferring control from one program to another, passing 255 parameters. Plus, **new extensions** to ACCEPT & DISPLAY verbs give better, faster interactive programming.



GIBSON MIX Benchmark Results

Calculated S-Profile
(Representative COBOL statement mix)
Execution time ratio

mbp COBOL	Level II** COBOL	R-M*** COBOL	Microsoft**** COBOL
1.00	4.08	5.98	6.18

128K system with hard disk required. *IBM/PC is an IBM TM; **Level II is a Micro Focus TM; ***A Ryan-McFarland TM; **** A Microsoft TM.

The complete COBOL.

An Interactive Symbolic Debug Package included standard; Multi-Keyed ISAM Structure; listing options allow source & object code, map & cross-reference checking; GSA Certification to ANSI '74 Level II; mbp has it all.

It's no surprise companies like Bechtel, Chase, Citicorp, Connecticut Mutual, and Sikorsky choose mbp COBOL; make it your choice, too. mbp is available at Vanpak Software Centers, or direct; just send the coupon, or call for complete information—today.

CIRCLE NO. 165 ON READER SERVICE CARD

mbp COBOL: 4 times faster.

mbp®

mbp Software & Systems Technology, Inc.

7700 Edgewater Drive, Suite 360, Oakland, CA 94621
Phone 415/632-1555

Please send complete mbp COBOL information to:

NAME ____________________

COMPANY ____________________

ADDRESS ____________________

CITY/STATE/ZIP ____________________

PHONE ____________________

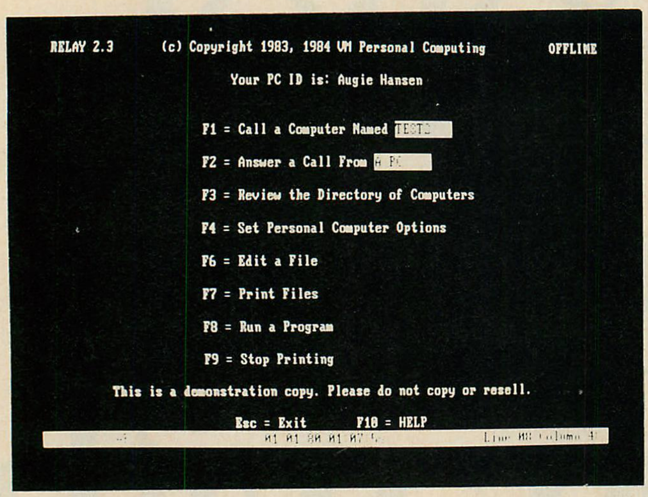


Photo 1: Relay's OFFLINE Menu

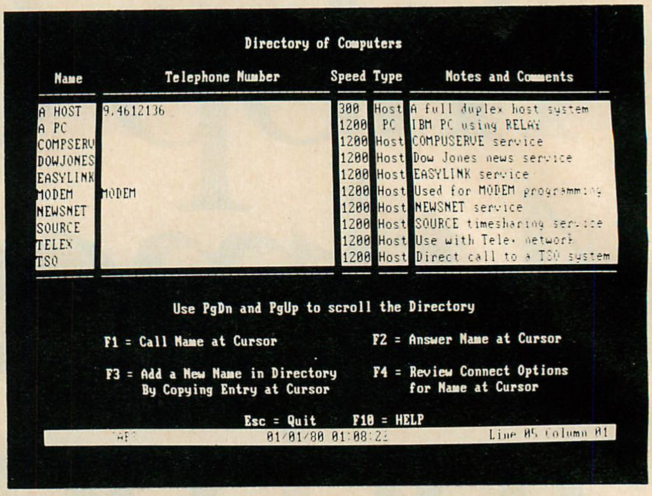


Photo 2: Selecting an Entry in Relay's Directory of Computers

include terminal emulation if the user is connected to a system that recognizes a VT100. This mode of operation is called the TTY mode.

When in the PC mode, Relay uses a proprietary packet-switching interface that has its roots in the X.25 standard. It is similar to the IBM Synchronous Data Link Control (SDLC) protocol, which is itself a subset of the High-level Data Link Control (HDLC) protocol defined in the CCITT-approved X.25 standard. The Relay protocol is designed to interleave simple messages with standard data blocks to effect full two-way simultaneous transmission on multiple virtual channels. From the user's perspective, several separate communications may be taking place concurrently.

Figure 1 is a high-level view of a typical PC-to-PC session using Relay. System A and system B are carrying on two very different kinds of communications at the same time: a "conversation" and a file transfer.

FORGET

EVERYTHING YOU THOUGHT YOU KNEW ABOUT PROGRAMMING IN BASIC

INTRODUCING BetterBASIC™

"It combines the best points of interpreted BASIC, Pascal, Forth, & Assembler...

...It's the first piece of software I'd spend my own money on."

Susan Glinert-Cole
Technical Editor,
PC Tech Journal

Dramatically improve your productivity with BetterBASIC.™

Introducing a new, breakthrough programming system called BetterBASIC.™ At last, the structure and functionality of advanced languages such as Pascal and "C" have been combined with the friendliness and ease of use of BASIC. BetterBASIC™ will dramatically increase the productivity of both new and experienced programmers. BetterBASIC™ is also a great Teaching Language.

Speed. Sieve of Erastosthenes Benchmark:
—BetterBASIC: 31.9 seconds.
—IBM PC BASIC: 191.1 seconds.

Support of large memory (to 640K).
Built-in WINDOWS support!

Each user may type on his or her keyboard to send "messages" to the distant system. Typed characters are echoed locally to the bottom half of the terminal screen. Each line of characters is formed into a message and sent to the other system to be displayed in the top half of the receiving system's terminal screen. Each user can view a window of text for each end of the link. When the windows fill up with text they are scrolled independently so that they show the latest transmitted and received messages.

While the conversation is continued, blocks of data are being transferred from system A's disk to a disk on system B. Each block of data has a virtual channel identification and a sequence number that allows the receiving system to put the data blocks back together correctly if they arrive out of order for any reason. Each virtual channel is actually bidirectional, so data can flow in either direction.

Relay places a higher priority on messages than on data packets, so the individual messages of a conversation can be shoved in between data packets. It is unlikely that anyone will type in messages fast enough to slow down the overall file transfer rate appreciably.

When a Relay-equipped PC is connected to a host that is running RELAY/VM or PC3270, proprietary protocols are used to handle the special requirements of each mode.

File Transfers

Relay accommodates all file transfer methods currently in vogue, from those that send one character at a time to those that send multiple blocks of data synchronously.

Files that contain only ASCII text may be sent in Relay's TTY mode using XON/XOFF flow control or sending one line at a time with a pause-proceed handshaking method that uses an assignable character (usually carriage return) as the turnaround character.

Binary files require more elaborate transfer techniques; several are provided. XMODEM is available in two flavors: the first matches the original CP/M description and uses a checksum error-detection scheme, and the second is a derivative of XMODEM now available on some bulletin boards. The second method uses the cyclical redundancy check (CRC) error-detection technique, which is able to detect compound bit errors that could fool a simple checksum method. The selected XMODEM option must be compatible with the version on the other end of the link for file transfers to be completed successfully.

The Relay protocol discussed earlier is provided for PC-to-PC communications in which both ends are equipped with Relay and for PC-to-mainframe communications in which Relay is present on the PC and Relay/VM is on the mainframe.

Good user feedback is provided on the message line at the bottom of the terminal screen and on a status screen that can be called up at any time to see how the communications session is progressing.

The Mainframe Connection

VMPC has obviously targeted the IBM mainframe market with Relay, and the product's 3270 mode hits the bull's eye for that market. Unlike most other 3270-series emulation products for the IBM PC, Relay requires no special hardware and only a companion software product for the mainframe side of the link. PC3270, sold separately, is a program that runs on the VM/370 SP operating system. It provides the interface that Relay needs in order to be able to simulate on the PC all of the functions of an IBM 3270-series terminal interface.

To set up for 3270 operation, it is necessary to select some connect options that set up Relay in the PC mode and to associate the options with a VM/CMS system name and

number in the directory of computers (this needs to be done only once for each VM/CMS system to be contacted). Once the software is installed and running, the PC will be able to display program menus and full screens of text for editing using mainframe programs in the same way a 3270-family controller and terminal would.

Relay/VM, which simulates a Relay-equipped PC-like interface on the host, is used to perform file transfers between a PC and a VM/CMS mainframe. When this software combination is running, any type of file may be uploaded or downloaded with Relay's error-checking and correction applied.

VT100 Terminal Emulation

VT100 terminal emulation is the weakest element of Relay. This mode is a recent addition, and it is not as complete as those of some competing programs (see "High-tech Mimicry," *PC Tech Journal*, September 1984, page 46, for a comparison of emulation programs).

Because the VT100 set-up screen is not implemented, it is not possible to toggle the margin bell on/off—it is permanently off. PC hardware limitations preclude such features as 132-column mode, double-height and double-width characters, smooth scrolling, and foreign-language character sets. VMPC has elected to leave out the LED display function and line wraparound control (always on). The desirable editing features (insertion and deletion of lines and characters) of the VT102/132 terminals that are supported by some VT100 emulators are not currently available in Relay.

Several popular mainframe visual text editors use the <Esc> key to perform special functions, such as exiting the input mode or forming special commands. Because <Esc> is used by Relay to mean "Go back to the previous context," and because it cannot be reassigned,

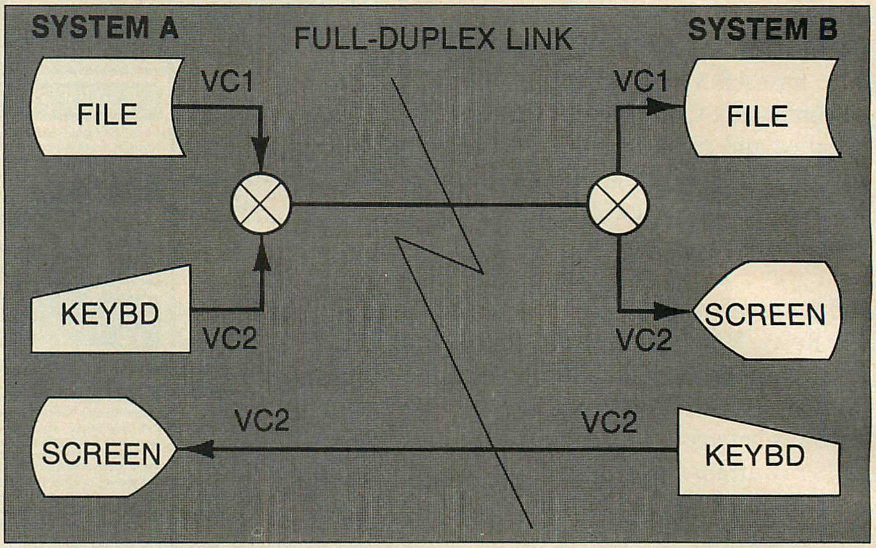


Figure 1: A High-level View of a PC-to-PC Session Using Relay

the <F2> key is used in the terminal mode to send the escape (decimal 27) code to the host.

I had trouble adapting to this juxtapositioning of keys and frequently found myself staring at a main on-line menu instead of the text I was editing. It is easy to recover from this (just type <F1>), but it's a nuisance nonetheless. Giving the user the ability to reassign the function to another key (as Crosstalk XVI's SWITCH function does, for example) would be a significant improvement.

Automatic Processing and Scripts

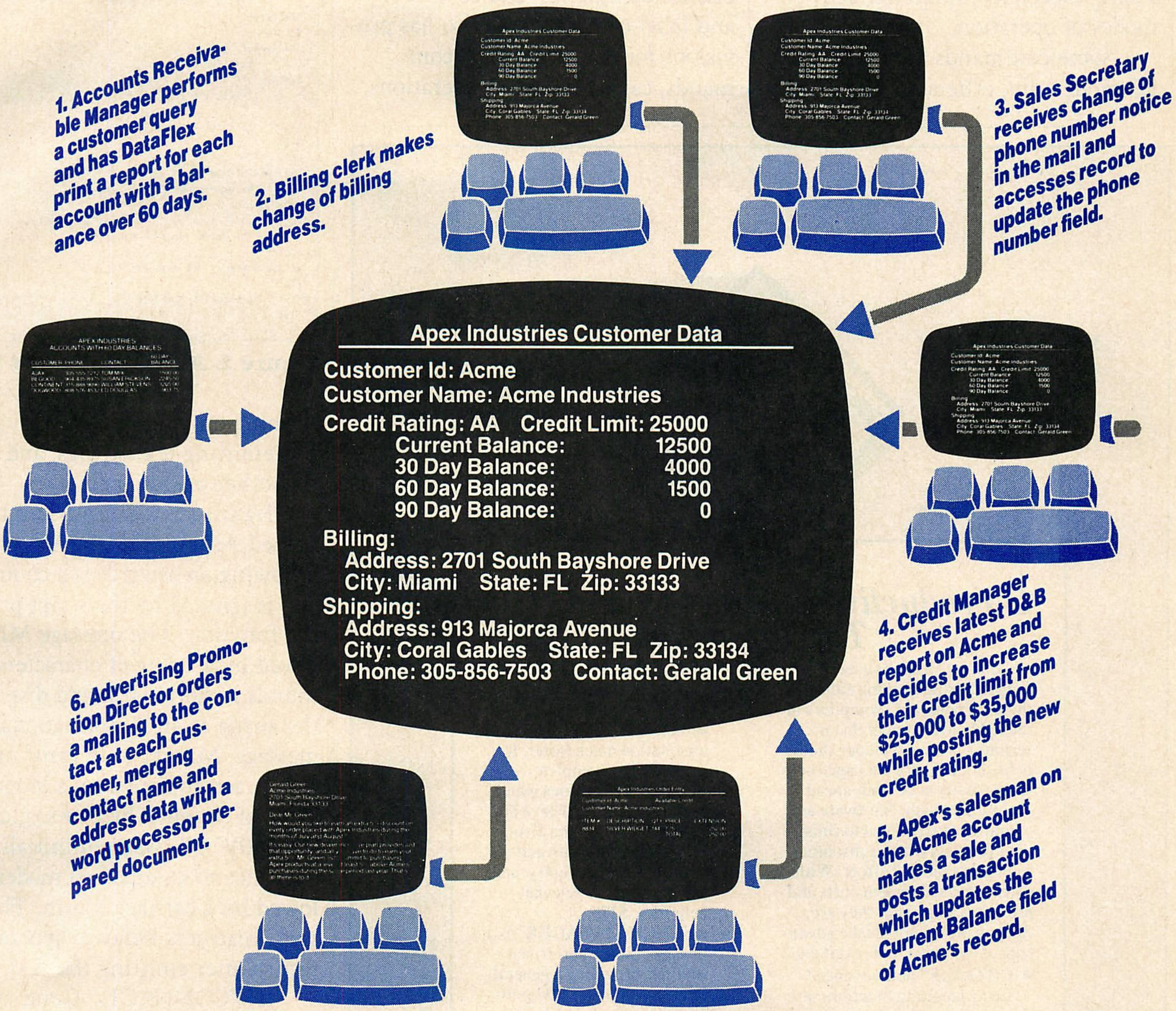
Automatic processing is a way of having an unattended PC that is running Relay perform tasks that would otherwise be done by a human operator. The tasks are the repetitive kind that usually have a pattern of known or expected exchanges, although conditional branching and comparison operators are available to handle considerable variation in conditions and data. Relay provides three levels of automatic processing that may be used individually or together:

1. DOS level—with DOS 2.0 (or later versions), batch files may be used to start Relay and, based on the return code when Relay exits, to run any DOS commands and other applications software. Relay can also use the redirection capabilities of DOS 2.0 (and later versions) to produce activity logs. In addition, a WAIT.COM command that can be used with any version of DOS is provided; when this command is used, Relay will wait for specified dates and times before running commands.
2. Relay level—Relay accepts command-line options to control its behavior in limited ways (how long to wait for a call, setting up an activity log). It also provides return codes that can be used by DOS batch file commands as described above.
3. Relay script level—files written in Relay Script Language that have an .SCR file-name extension may be used to perform a wide variety of tasks, such as automatic log-on and log-off sequences, automated file transfers, and other functions. Functions packaged in script files may be used as modules called from other scripts to automate Relay communications sessions fully.

Because Relay is a menu-controlled program, it does not have an English-language command syntax of the kind used by Crosstalk XVI. The Relay Script Language tends to

ALL AT ONCE!

AND NEVER A "LOCKED OUT" USER!



DataFlex is the only application development database which ***automatically*** gives you true multi-user capabilities. Other systems can lock you out of records or entire files for the full time they are being used by someone else. DataFlex, however, locks only the data being changed, and ***only*** during the micro-seconds it takes to actually write it to the file! The updated record is then immediately available. The number of users who can access, and change, records at the same time is limited only by the number of terminals on your system or network. Call or write today for all the details on DataFlex... the true multi-user database.

DATAFLEX™

8525 SW 129 Terrace, Miami, FL 33156 (305) 238-0012
Telex 469021 DATA ACCESS CI

CIRCLE NO. 146 ON READER SERVICE CARD

Compatible with CP/M-80, MSDOS networks, MP/M-86, Novell Sharenet, PC-Net, DMS Hi-net, TurboDOS multi-user, Molecular N-Star, Televideo MmmOST, Action DPC/OS, IBM PC w/Corvus, OMNINET, 3Com EtherSeries and Micromation M/NET.
MSDOS is a trademark of Microsoft. CP/M and MP/M are trademarks of Digital Research.

be terse and not particularly mnemonic, which made it a chore to write a script. Using this language takes practice and ready access to the user's guide. Once learned, however, it brings considerable power to Relay, especially in unattended modes of operation.

Some commands are usable in PC mode only, others are usable in TTY mode only, and many are available in both modes. The language provides conditional and absolute branching, comparison operators within conditional tests, a variety of screen, keyboard, and port control commands, and comment and label commands. It also has provisions for executing DOS commands, canceling active operations,

```
*
*  RELAY 2.3 - Sample ANSWER script for Hayes Smartmodem
*
*  (c) Copyright 1983, 84 by VM Personal Computing, Inc.
*
*  This script demonstrates RELAY's support for general
*  "intelligent" modems by calling via a Hayes Smartmodem.
*  Of course, RELAY's native Smartmodem support is more
*  advanced; you should use this script only as a sample.
*
MIWaiting to answer the telephone ... Please stand by.
*       First, prepare the modem ...
CATE0V1Q0
W3COK
IFW Gerror
*       Wait for RING reply ...
WCRING
*       Answer the phone ...
CATA
*       Wait for the result line ...
W10C
W3C
IFS/CONNECT/ Gend
*       No carrier or similar error ...
MENo Carrier on line!
FX2
*       The modem is not responding ...
-error
MEYour modem is not responding!
FX1
*       Connection complete ...
-end
```

Figure 2: A Sample Script File

running other scripts, displaying and emitting text strings, and myriad other operations.

Figure 2 is a sample script file named ANSWER.SCR. Anything following an asterisk is a comment and is ignored by the script-processing function. The message MI followed by a string of characters sounds a short alarm and displays the string as an "informational" message. MEtext is an error message that is accompanied by a longer burst of sound. The command CATE0V1Q0 issues an initialization sequence to an attached modem, followed by a carriage return. The letter *C* instructs Relay to append the return after emitting the text string.

IFS/CONNECT/ Gend is a conditional expression that checks to see if the result of the previous wait command (W3C) was the string CONNECT. If the correct string was received, the Gend command, which means GOTO the label "–end," is executed. Otherwise, an error message is displayed and a warning tone sounded. Considerable flexibility and power is available through the use of scripts.

Unattended Answer Mode

Relay may be set up to run unattended and to answer calling sys-

Now, X-shell™ brings Unix™ capability to your PC for an amazingly low $295

Computers. You know at least two things about them: One, they should be a tool for solving your problems. Not creating them. Two, you want to spend as little time as possible on one.

X-shell Software Tools Package

That's why Standard DataCom, Inc. developed X-shell, a command interpreter software tools package for IBM PCs and MS-DOS/PC-DOS operating systems.

X-shell helps your computer solve your problems faster. Makes the time you spend on the computer more efficient. Squeezes more out of your computer investment.

Proven Unix Features

X-shell gives you Unix capability by providing these proven Unix features:

1. Support for pipelines and filters
2. Input & output redirection
3. Variable substitutions
4. Command substitutions
5. Filename expansion
6. Nested command files
7. Structured flow control: if/then/else, while/do/done, until/do/done, case/esac, for/do/done

Unix is a trademark of Bell Laboratories
X-shell is a trademark of Standard DataCom, Inc.

Hardware Requirements (where you can save hard cash)

IBM PC or XT (or compatible)
256K bytes of RAM
2 Disk Drives—Hard or Floppy

Since X-shell can be run using floppy disk drives, you can save a bundle of money on your hardware. Hard disks are expensive. Floppies aren't. It's that simple.

Software Requirements

PC-DOS version 1.1 or 2.0
Software Disk Emulator

Since X-shell runs on PC-DOS, it supports existing PC-DOS software.

Over 40 Commands

X-shell's commands include:

basename	–strip extension from file name	num	–number lines
cat	–concatenate files	pr	–format files for printing
cd	–change directory	print	–pr directed to printer
clear	–clear monitor screen	pwd	–print working directory
cmp	–compare files	rm	–remove files (delete)
comm	–output lines common to two files	sh	–shell (command interpreter)
cp	–copy files	size	–size of object code
cpio	–file backup/archival	sort	–sort numerically or alphabetically
date	–get or set date and time	sum	–checksum file
echo	–echo arguments to stdout	tail	–output last lines of file
expand	–expand tabs into spaces	tee	–pipe fitting
expr	–string and arithmetic evaluation	test	–test file's or string's characteristics
false	–do nothing, unsuccessfully	time	–determine time to execute a command
find	–produce list of selected files	tr	–translate or delete characters
grep	–search files for specified pattern	true	–do nothing, successfully
hd	–hex file dumper	unexpand	–replace spaces with tabs
head	–output 1st lines of file	uniq	–remove duplicate lines
ls	–sorted directory list	wc	–count chars, words and lines
more	–copy files to display	words	–output file 1 word per line
mv	–move files (rename)		

Dealer Inquiries Invited

What You Get For $295.

1. Complete Documentation, including: Installation, Tutorial and Reference Manuals.
2. Complete X-shell Software Tools Package on two diskettes.
3. A more efficient use of your computer time.
4. More effective use of your PC.

How To Order X-shell

Get the software package that gives you Unix capability for an amazingly low $295.

Call Standard DataCom, Inc. 415-775-8882 with your Visa or MasterCard order, or for more information.

Or ask your dealer.

Standard DataCom, Inc.

X-shell™

Ask us about X.25

Standard DataCom, Inc.,
1550 California Street #6195,
San Francisco, CA 94109

CIRCLE NO. 299 ON READER SERVICE CARD

tems that also are running Relay. A file containing special instructions is created to control the way Relay interacts with callers in this mode. Multiple levels of access to system disks provide flexible operating conditions for a wide range of callers. The file may contain some or all of the following items:

LOGMSG=logon message filename (the file may contain any message that all callers should see when they log in)

HELP=help message filename (the file to be displayed when a caller asks for help by pressing the <f10>) key

PW=password (a general log-on password that allows callers to read and write whatever disks are specified upon entry to the unattended mode)

MASTER=password (allows access to all disks regardless of access limitations imposed on general callers)

READONLY=password (allows callers to read from but not write on the disks that are made available by the access level specified)

Relay also will keep a printed record of all transactions and commands during unattended mode if the printer is on and if the user has not disabled the logging option.

FULL SCREEN EDITOR

In addition to all of the sophisticated communications functions in Relay, the product also has a decent full screen editor. The file being edited must reside completely in memory, along with DOS, the communications functions, and anything else that's memory-resident. The editor is essentially command-driven, although the function keys duplicate some of the available editing commands.

The display is a viewport onto a text scroll that is up to 240 characters wide and as many lines long as available memory will permit. The viewport may be moved right or left in increments of 80 columns, and the cursor may be moved anywhere on the screen by using the arrow keys. It may take some practice to use this editor effectively, because the screen does not scroll up when the user tries to move the cursor off the bottom of the screen; instead, the cursor wraps back to the top line. Other commands scroll the screen and page up and down through the file, moving a screen or half-screen at a time. Photo 3 shows a sample editing session in progress.

EXECUTING COMMANDS

To use another editor rather than the one provided, the user can escape to the ONLINE menu and use the "Run Program" function—<F8>—to call in the preferred program, providing there is enough memory available. Photo 4 is an example of how to execute a DOS 2.0 (or later version) resident command (such as TYPE, DIR, etc.).

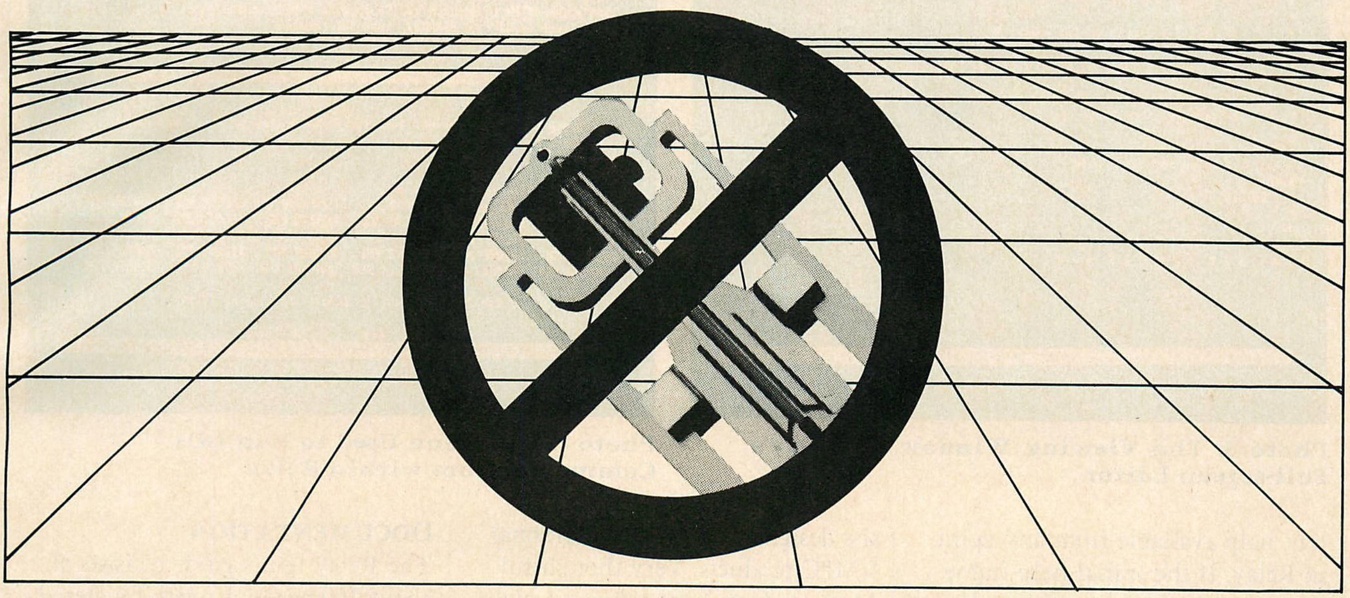
In fact, pressing <F8> from either the ONLINE or OFFLINE menus lets the user execute most command/program files (.COM and .EXE). This function worked perfectly with all of the text editors I tried (EDIX, VEDIT, and WordStar in the nondocument mode), with VisiCalc (a supplied program called VL80.COM must be run first), and with numerous other applications.

ON-LINE HELP

The help feature of Relay is one of the most extensive available in a microcomputer product of this type. Help frames are context sensitive and are actually helpful. The user's guide claims that once a user is familiar with the help system, he will seldom need to refer to the guide. It's true. I was able to find appropriate and complete information for nearly every situation encountered in the use of Relay.

Photo 5 shows the detailed and relevant help provided during the modem-selection activity of initialization. It is indicative of the excel-

No Head is Better than One!



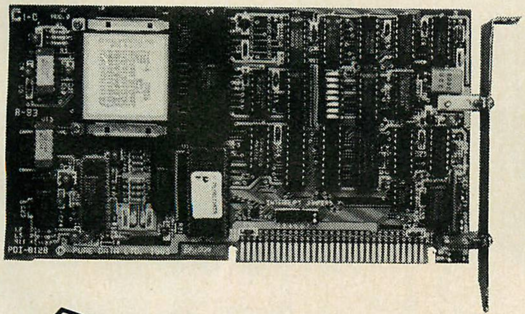
Bubble Storage for the PC and PC XT.

Bubble memory combines the best features of disk and solid state memory with extreme reliability. Unlike a floppy disk, it's entirely solid state with no moving parts. This makes it impervious to dust, dirt, fumes, and vibration. And, unlike normal memory, it doesn't lose data when the power goes off.

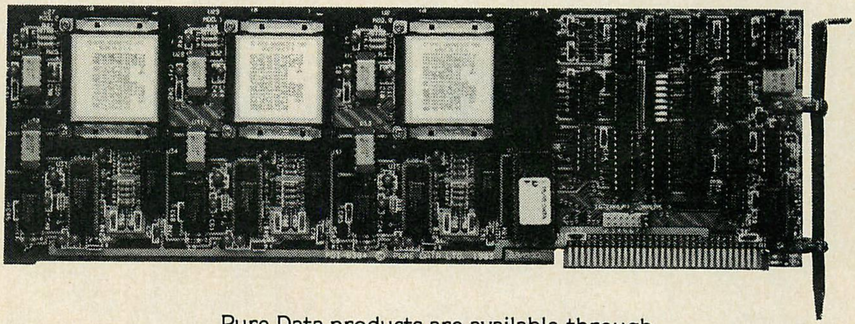
Now you can enjoy all the advantages of bubble memory combined with the famous Pure Data quality and reliability for your IBM PC or XT.

- PDIB-128 provides 128 KBytes in one slot
- PDIB-384 provides 384 KBytes in one slot
- Not affected by power failure
- Faster than a floppy disk
- Extremely reliable
- Standard DOS 2.0 disk-type device
- Compatible with all DOS software
- No patching of any system files
- Password option for computer and/or bubble
- Password cannot be bypassed by software
- Comprehensive diagnostics and other utilities
- Fully illustrated installation and operation manual
- Technical support hot-line
- Guaranteed 48 hour service
- Bubble uses 2 I/O addresses and no memory space
- Interrupts are supported but not required
- DMA is supported but not required
- On-board EPROM socket
- Pure Data quality and reliability
- No moving parts

PDIB-128



PDIB-384



Pure Data Ltd

860 Denison Street,
Markham, Ontario, Canada L3R 4H1
(416) 475-2424 (416) 498-1616

Pure Data products are available through.

OnTrak DATA SYSTEMS
1335 Valwood Parkway, Suite 108,
Carrollton, Texas 75006
Telephone (214) 620-8000
P.O. Box 815155, Dallas, Texas 75381

Highland Computer Corporation, Keysoft International, Computerland, Computer Innovations, Compugroup, ECOSEA Technologies and others.

CIRCLE NO. 190 ON READER SERVICE CARD

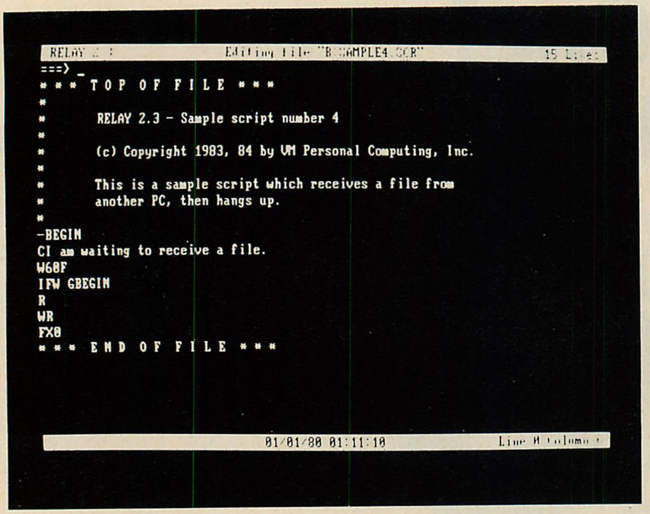


Photo 3: The Viewing Window of Relay's Full-screen Editor

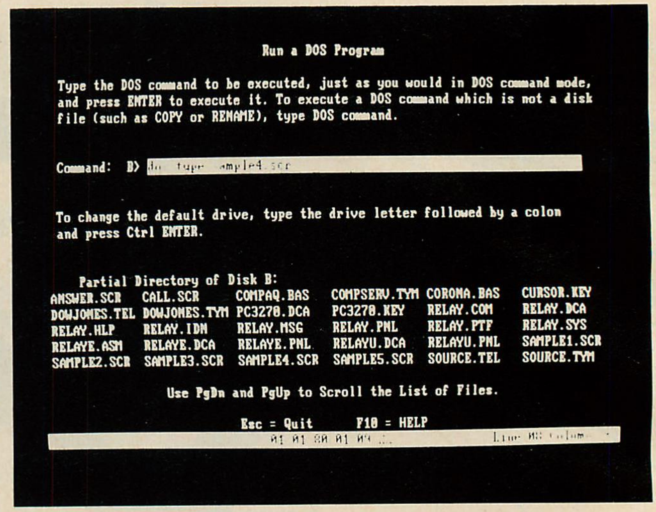


Photo 4: The Menu Used to Run DOS Commands from within Relay

lent help available from any point in Relay. If the initial help frame doesn't give adequate information, more detailed ones are accessible by typing <PgDn>.

Relay even has an on-line feedback form that can be printed and filled in to report bugs, congratulate the designers, and order additional VMPC products. Very thoughtful. Most products have either no such forms or only one printed copy in the manual. VMPC gives you an electronic master that can be printed out as often as necessary. And they encourage its use.

DOCUMENTATION

The Relay user's guide consists of about 250 pages of typeset material in a half-size ring binder. The coverage is complete and well organized, and the writing is clear and well illustrated. The index is accurate but not complete enough.

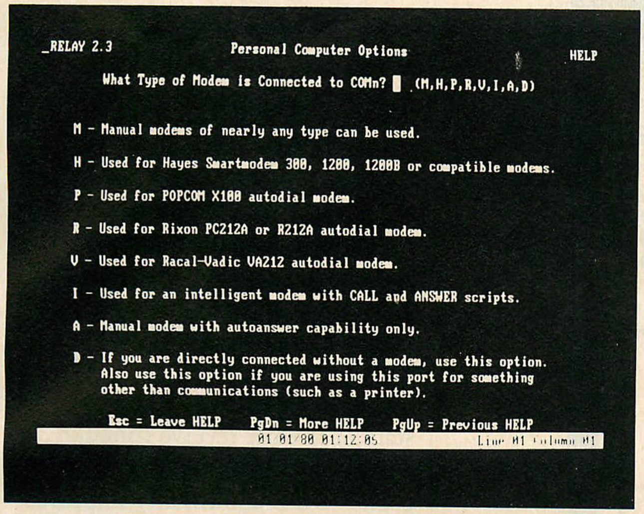


Photo 5: A Help Frame Called from the Modem-selection Menu

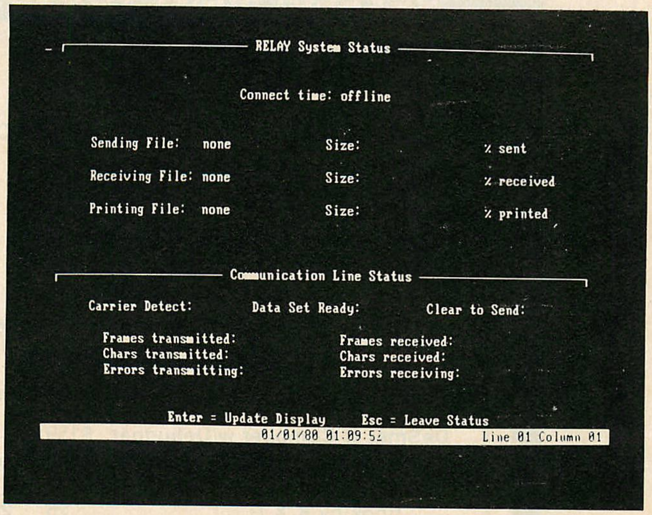


Photo 6: The Relay System Status Screen

Much of the material covered in the user's guide is also available on-line, except for the examples and illustrations. It was easier on occasion to find what I needed by going into the program and pressing the help key, <F10>. That will not work for the programs that are supplied with Relay in separate files. It's a good idea to read through the manual once to get an idea of the abilities of the product.

SUMMARY

This program makes me question my long-standing uneasiness about menus, probably because Relay's menu tree is rather broad and flat. It is never necessary to descend more than one or two levels to get to a place where work can be done, and it is possible to jump directly to the status screen (photo 6) and help frames at any time.

Data capture to a printer is supported in the usual ways—pressing <Ctrl>-<PrtSc> toggles the printer ON/OFF, and pressing Shift-<PrtSc> prints the current contents of screen memory. The Relay package supports data capture directly to disk using its file-receive function. After specifying the receiving file, the user may toggle the data capture ON/OFF by pressing <F4>. A message on the 25th line of the screen indicates the current state of the file-receive function.

User support is reportedly very good. VMPC continually improves and expands its products and provides an on-line capability for registered users to obtain the latest patches, called PTFs (program temporary fix). These are downloadable programs that, when executed, modify the Relay load module, RELAY.COM, to incorporate bug fixes and program upgrades; these programs also update a disk file called RELAY.PTF that contains the latest list of applied PTFs.

Users who want a reasonably priced, feature-packed communications package for their computers should give this one a serious look. The more I use it, the better I like it—it's proving to be a real productivity tool. I'll like it even more when the quality of the terminal-emulation feature is on a par with the rest of the program.

Relay, Version 2.3
VM Personal Computing, Inc.
475 5th Avenue, Room 411
New York, NY 10017
212-686-1450
Language: Macro Assembly Language
List: $149

CIRCLE 491 ON READER SERVICE CARD

References

Loomis, Mary E. S., *Data Communications*, Prentice-Hall, 1983.

McNamara, John E., *Technical Aspects of DATA Communication*, Digital Press, 1977.

X.25, "Computer Communication Review," vol. 10, nos. 1 and 2 (January/April 1980).

PC LINK

INDEPENDENT STORE IN NYC DEDICATED TO IBM® PC SOFTWARE AND PERIPHERALS ONLY

BY APPOINTMENT WE DEMONSTRATE MOST OF THE PRODUCTS WE SELL.

STORE HOURS: 9:00 A.M.–5:30 P.M. E.S.T. MON.–FRI.
Saturdays 10 A.M.–4 P.M.
Terms, conditions and prices differ in our store.

PC LINK, CORP. 29 WEST 38TH ST. 2ND FL., NEW YORK, NY 10018

SPECIALS OF THE MONTH

CIRCLE NO. 183 ON READER SERVICE CARD

AST MEGAPLUS II	MICROPRO	BROTHER
Clock/Serial/64K 259.00	WORD STAR Professional ... 269.00	HR-25 .... 659.00

BUSINESS SOFTWARE

ALPHA SOFTWARE	
DATA BASE MANAGER II	$205.00
EXECUTIVE PACKAGE	109.00
TYPE FACES	89.00
ASHTON–TATE	
dBASE III	CALL
FRAME WORK	CALL
CONTINENTAL	
HOME ACCOUNTANT PLUS	89.00
TAX ADVANTAGE	45.00
PROPERTY MANAGEMENT	319.00
ENERTRONICS	
ENERGRAPHICS	CALL
FOX & GELLER	
D. GRAPH/QUICKCODE/DIUTIL	CALL
OZ	CALL
FRIENDLYSOFT	
FRIENDLY WRITER	60.00
HARVARD SOFTWARE	
HARVARD PROJECT MANAGER	CALL
HAYES	
SMARTCOM II	95.00
INNOVATIVE SOFTWARE	
TIM IV	CALL
I U S	
ACCOUNTING AR/AP/GL ea.	CALL
INVENTORY CONTROL	CALL
ORDER ENTRY	CALL
PAYROLL	CALL
EASY WRITER II/SHELLER/MAIL	199.00
EASY FILER	199.00
EASY PLANNER	159.00
LOTUS DEVELOPMENT CORP.	
LOTUS 1-2-3. (No Mail Or Phone Orders)	CALL
SYMPHONY (No Mail Or Phone Orders)	CALL
MICRO DATA BASE	
KNOWLEDGE MAN	CALL
MICRO PRO	
INFOSTAR	CALL
WORDSTAR PROFESSIONAL	SEE SPECIAL
MICRORIM	
R BASE 4000	309.00
EXTENDED REPORT WRITER	109.00
CLOUT (RE8.256K)	139.00
PROGRAM INTERFACE	295.00
MICROSOFT	
MULTIPLAN	139.00
WORD	CALL
COMPILERS	CALL
MICROSTUFF	
CROSSTALK	105.00
NORTH AMERICAN BUSINESS	
MEMORY SHIFT	69.00
NORTON COMPUTER	
NORTON UTILITIES	55.00
PC SOFTWARE	
CREATABASE	65.00
PEARL SOFT	
PERSONAL PEARL	CALL
ROSE SOFT	
PROKEY 3.0	85.00
STSC	
APL*PLUS	CALL
SATELLITE SOFTWARE	
WORD PERFECT	269.00
SAWHNEY SOFTWARE	
TAX MODE	CALL
PLAN MODE	CALL
SOFTWARE ARTS	
TK SOLVER	249.00
FINANCIAL MANAGEMENT	79.00
MECHANICAL ENGINEERING	79.00
SOFTWARE PUBLISHERS	
PFS SERIES	CALL
SOFTWARE SYSTEMS INC.	
MULTIMATE	279.00
WARNER SOFTWARE	
THE DESK ORGANIZER	209.00

DISK DRIVES

IOMEGA BERNOULLI BOX	CALL
QUME T142	CALL
SYSGEN	CALL
TANDON TM 100-2	200.00

MULTIFUNCTION BOARDS

64 RAM CHIPS	55.00
AST CARDS	CALL
QUADBOARD CARD	CALL
AMDEK MAI CARD	CALL
PC MOUSE	CALL
FTG LIGHTPEN	CALL
PLANTRONICS COLOR PLUS CARD	CALL
HERCULES	CALL
HELIX BUBBLE MEMORY	CALL

ENTERTAINMENT & EDUCATION

ACROSS THE USA	25.00
ATI TRAINING PACKAGES ea.	55.00
ATTACK ON ALTAIR	29.00
BIG TOP	27.00
CONQUEST	29.00
COSMIC CRUSADERS	27.00
DEADLINE	37.00
DIGGER	29.00
THE EXTERMINATOR	29.00
FLIGHT SIMULATOR–NEW VERSION	35.00
FRIENDLY ARCADE	29.95
FRIENDLY PC INTRO SET	29.95
FROGGER	25.00
GORGON	29.00
HI RES #4	27.00
KID STUFF	32.00
MASTER MINER	27.00
MOON BUGS	29.00
MY LETTERS, NUMBERS & WORDS	32.00
PC CRAYON	39.00
PC TUTOR	49.00
PLANET FALL	39.00
SERPENTINE	25.00
SNACK ATTACK	27.00
SPINNAKER SERIES	CALL
STYX	29.00
SUSPENDED	39.00
WITNESS	39.00
WIZARDRY	39.00
ZORK I, II, III ea.	27.00
ZURAN DEFENDER	25.00

ACCESSORIES

CURTIS PRODUCTS	
DISPLAY PEDESTAL	35.00
MONO CABLE	38.00
KEYBOARD CABLE	30.00
SYSTEM STAND	17.00
COLOR CABLE	45.00
SURGE PROTECTORS	CALL
KRAFT JOYSTICK	39.00
HAYES MACH III JOYSTICKS	45.00
KEYBOARD DRAWER	75.00

MONITORS

AMDEK 310A	$175.00
COLOR II+	CALL
PRINCETON GRAPHIC HX-12	479.00
SR-12	CALL
MAX-12	179.00
ZENITH RGB MONITOR	479.00

MODEMS

HAYES SMART MODEM	$205.00
HAYES SMART MODEM 1200	489.00
HAYES SMART MODEM 1200B	449.00

DISKETTES

DYSAN	
S.S./D.D.	$24.00
D.S./D.D.	34.00
MAXELL	
S.S./S.D.	23.00
D.S./D.D.	33.00
VERBATIM (5 year warranty)	
21144 HEAD/CLEAN KIT	10.50
S.S./D.D.	22.00
D.S./D.D.	32.00
WABASH (Lifetime Warranty)	
D.S./D.D.	25.00

PRINTERS

DIABLO 630	$1699.00
DIABLO 630 TRACTOR FEED	249.00
DYNAX DX-15	CALL
IDS PRISM 132 COMPLETE	1499.00
OKIDATA 92P	CALL
93P	CALL
84P	CALL
TOSHIBA P1351	1289.00
EPSON FX SERIES	CALL
NEC 3550	1599.00
PRINTER CABLE/PARALLEL	25.00

CALL TOLL FREE 800-221-0343 All prices listed in this ad are effective Oct. 1-31, 1984. All Brands are Registered Trademarks. IBM is a Registered Trademark of IBM Corp. **IN NEW YORK CALL 1-212-730-8036**

TERMS AND CONDITIONS

We reserve the right to repair, replace or return to manufacturer for repair, all goods acknowledged faulty or damaged on receipt by customer. ***Customer must call for Return Authorization Number before returning any goods.*** Prompt attention will be given to all damaged and faulty returned goods. Any goods returned for credit are subject to 10% restocking charge, plus shipping charge. No returns for credit on any software. Customer must deal with the manufacturer directly if the customer finds any false claims made by the manufacturer. All goods are shipped U.P.S. only. Add 2% of price, or minimum of $3.00 for shipping. Shipping charges on Printers & Accessories vary. We do not ship C.O.D. Please allow one to two weeks for personal or corporate checks to clear. To expedite shipping send money order, certified cashier's check, or charge to your **VISA, MasterCard, WE DO NOT Add a Service Charge For Credit Card Usage.** Prices subject to change without notice. NOT RESPONSIBLE FOR TYPOGRAPHICAL ERRORS.

THERE'S NO

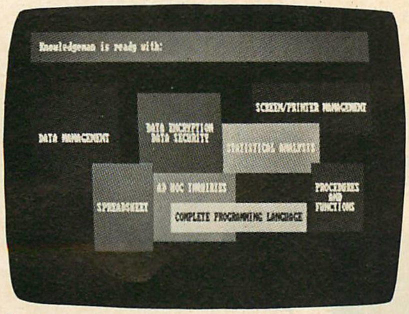
KnowledgeMan™ and You. The possibilities are endless.

To succeed in business, you need a partner that's fast, flexible, intelligent and easy to work with. A partner that can help turn your big ideas into well-conceived reality. One that gives you the support you need to make critical decisions confidently.

No partner can give you more of what you need than KnowledgeMan, the knowledge management software from MDBS.

A powerful partner.

KnowledgeMan helps you manage more knowledge, in more ways, than ordinary software. It can help you make better decisions on just about everything from production scheduling to financial planning to market



forecasting. KnowledgeMan and its optional components offer data management, spreadsheet analysis, statistical analysis, text processing, forms management, business graphics, programming and more.

The key to KnowledgeMan's versatility is its exclusive synergistic integration, allowing you to accomplish your computing needs within one program. Unlike other software, there's no need to exit one function before entering another. The result: different kinds of processing can be intermingled. Quickly and easily.

A partner that speaks your language.

For all of its power and sophistication, KnowledgeMan is remarkably simple to understand. Even a beginner can start putting KnowledgeMan to work in minutes. With a single query, you can obtain related data from unlimited multiple tables. You can even teach KnowledgeMan to understand your own jargon.

A partner that helps you along.

The on-line HELP facility allows you to draw on 6800 lines of helpful information organized into 380 screens. If you have a problem or question, KnowledgeMan allows you to access the pertinent HELP screen immediately. Each screen is carefully designed to provide a quick reference guide to KnowledgeMan commands.

A partner that gives you room to grow.

Ordinary software packages can be frustratingly easy to outgrow. Not KnowledgeMan. Each KnowledgeMan component has more power than you'll probably ever need—far more than conventional integrated programs. With KnowledgeMan, you don't sacrifice capability, capacity or convenience. So with KnowledgeMan, you spend your time solving problems—not trying to overcome software limitations.

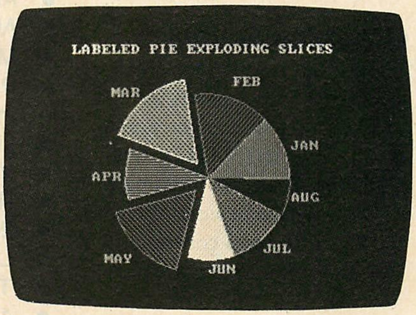
A partner that protects your interests.

KnowledgeMan offers sophisticated security features. Unauthorized access to data is next to impossible, thanks to password checking, thousands of access code combinations and data encryption.

So your secrets are safe with KnowledgeMan.

A partner you can build on.

To add yet another dimension to KnowledgeMan's capabilities, you can get fully-integrated options like K-Graph, an extensive business graphics facility that



lets you plot information in a variety of colorful graphs, charts and diagrams. For text processing, the K-Text option lets you incorporate data into written documents quickly and easily. Or, create highly-polished, full-color customized forms with K-Paint, our forms painting option. To short-cut the keyboard, put the K-Mouse option to work.

A partner you should get to know better.

To see KnowledgeMan in action, visit your dealer. Or contact Micro Data Base Systems, Inc., P.O. Box 248, Lafayette, IN 47902, (317) 463-2581, Telex: 209147 ISE UR.

It may be the beginning of a long, successful partnership.

Current version is 1.07 as of 9/10/84.
KnowledgeMan, K-Graph, K-Paint, K-Text, and K-Mouse are trademarks of Micro Data Base Systems, Inc. MDBS is a registered trademark of Micro Data Base Systems, Inc.

KNOWLEDGE man™

The Knowledge Management Software from MDBS

Operating Systems: PCDOS, MSDOS, CP/M-86. Minimum RAM required: 192K, K-Graph: PCDOS only.

CIRCLE NO. 219 ON READER SERVICE CARD

What do IBM, HP, Tektronix, and DEC have in common?

TBASIC!

TBASIC. A new, portable, incrementally compiled language with integral GPIB and Graphic statements compatible with Tektronix 4050A BASIC.

Graphics Galore

TBASIC gives you over 60 high level graphic commands for a variety of graphic computers and terminals. From IBM PC to high resolution Tektronix terminals, TBASIC lets you access their intelligent features consistently and with a minimum of effort.

GPIB Genius

Instrument control and operation of shared peripherals does not need to be awkward. TBASIC provides syntax for both simple access of GPIB devices through normal I/O commands that understand GPIB, and special purpose functions for sophisticated instrument control.

For Professionals and Part-timers

TBASIC is an excellent development language for general purpose or technical programs. It is extremely easy to learn and use with features that professional and novice programmers alike will find useful for applications ranging from business to scientific.

TBASIC Triumphs

No other high level language gives you complete access to and control over all elements of your computer system like TBASIC does. No other language gives you so many advanced and useful features.

Features Fantastic

- Graphics: over 60 functions and commands
- GPIB access is incorporated into standard I/O statements
- Tektronix 4050A BASIC programs are compatible
- Incremental compilation produces fast intermediate code
- Interactive nature of BASIC is fully preserved
- Automatic line numbering with optional line increment
- Abbreviated line entry format (3 letters/keyword)
- Multiple statements per line are supported
- Program listings are neatly formatted with indented loops
- Instant syntax checking "shows" you errors
- Line editor with 7 function keys
- Variable names can be up to 31 characters long
- Rename variables in programs without editing
- 20 special function "user definable keys"
- Line labels are supported for structured programming
- True subprograms with local or global variables
- 15 digit precise decimal floating point arithmetic
- Integer data types conserve memory and speed execution
- 8087 math coprocessor supported
- All math functions can operate on arrays
- Matrix operations are included
- Boolean, logical, and binary operators are supported
- Relational operators allowed in math expressions
- Device independent I/O scheme
- Random and Sequential file types
- INPUT statements allow prompts on same line
- Debug facilities include numerous trace options
- Single step function key for manual program execution
- Complete error trapping and handling is provided

TBASIC

Superior Support

Computers:
IBM PC/XT, HP-150, Tektronix 4100 Series, DEC VAX, and others

Operating Systems:
MS-DOS, PC-DOS, CPM, UNIX, VMS,...

GPIB Interface Cards:
National Instruments, Ziatech,...

Graphic Cards:
IBM, Hercules,...

Graphic Terminals:
Tektronix, Seiko,...

Plotters:
Cal-Comp, Hewlett-Packard, Houston Instruments, Tektronix,...

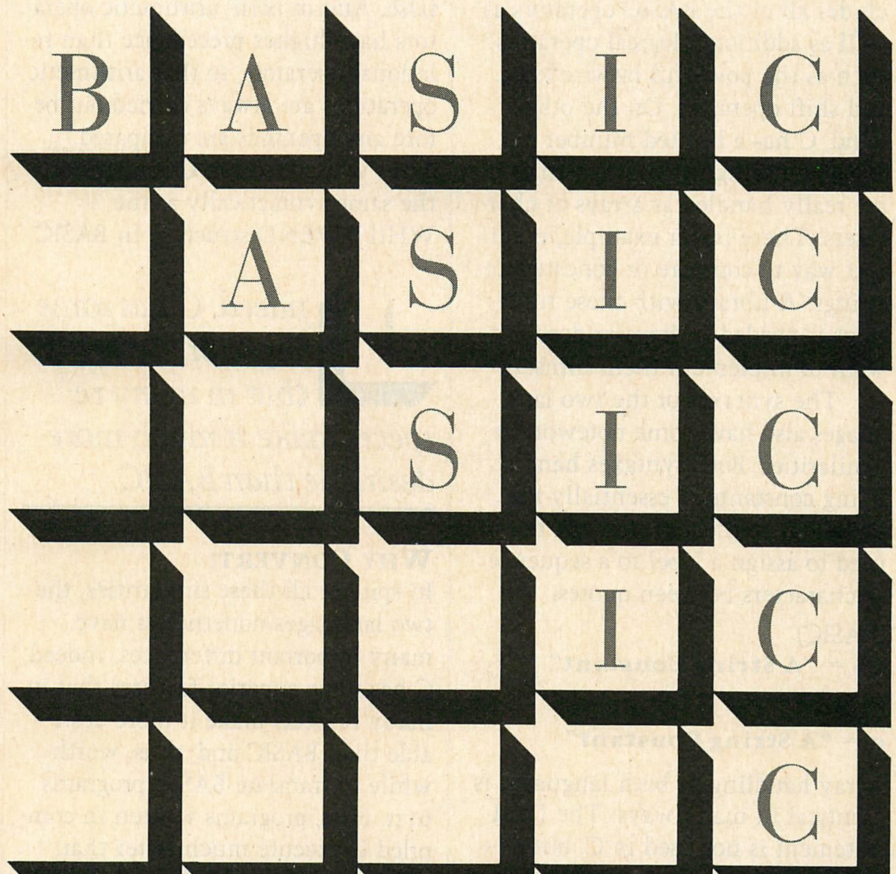
These and many other popular computers, operating systems, and peripherals are, or will be supported by TBASIC.

IBM PC and other micro versions of TBASIC are priced at $795.
A supermini version of TBASIC is available for DEC VAX at $2000.
Call 1-801-224-6550 to order your copy of TBASIC now.
Further information and demonstrations are available upon request.
TransEra Corporation
3707 North Canyon Road
Provo, Utah 84604

TransEra

CIRCLE NO. 181 ON READER SERVICE CARD

ERNEST TELLO



BASIC TO C

Two products that ease the transition from BASIC to C

Largely because of its greater portability, speed, and power, C is rapidly replacing BASIC as the favored language for many new software products. Some software developers are translating their programs from BASIC into C. A number of tools have emerged to help in the crossover between BASIC and C. Two of these tools—BASIC_C, a library of BASIC functions for several popular C compilers, and BASTOC, a BASIC to C translator that produces C source from BASIC text files—are evaluated in this article.

BASIC vs. C

At first glance, the differences rather than the similarities between C and BASIC are the most obvious. C is always compiled; BASIC is generally an interpreted language. C is structured; BASIC is not. C supports user-defined multi-lined functions, while BASIC is limited to a fixed vocabulary. C usually adheres strictly to the lower-case format, whereas in BASIC, which accepts both upper and lower case, reserved words usually appear in upper case.

C has a far richer set of operators, allowing for its well-known ability to express complex logical

Ernest Tello is the cofounder and director of research and development of a consulting and software firm called Integral Systems. He has a master's degree in engineering science from Rensselaer Polytechnic Institute.

operations very concisely. C includes all of the BASIC operators as well as additional logical operators, such as the powerful bitwise logic and shift operators. On the other hand, C has a limited number of string-handling functions: strings are really handled as arrays of characters. There is, for example, no direct way to compare or concatenate strings. A library with these functions is needed if the user does not want to implement them himself.

The syntaxes of the two languages also have some noteworthy similarities. Both syntaxes handle string constants in essentially the same way. An equal sign can be used to assign a label to a sequence of characters between quotes:

(BASIC)
Q$ = "A String Constant"

(C)
q = "A String Constant"

Array handling in both languages is identical in many ways. The DIM statement is not used in C, but declaring arrays and accessing array elements use essentially the same conventions of parentheses and indexing numbers.

Array declarations:
(BASIC)
DIM A(1,2,3,4)
DIM A$(7)

(C)
float a[1][2][3][4];
char a[7];

Accessing first array element:
(BASIC)
A(0,0,0,0)
A$ (0)

(C)
a[0][0][0][0]
a[0]

The same syntax is used for initializing and assigning variables, and single-precision floating-point numbers appear the same in both languages. The arithmetic operators, except for a few minor exceptions, are essentially identical. In both BASIC and C any nonzero operand is considered true; a zero operand is false. And in both, arithmetic operators have higher precedence than relational operators, so that arithmetic operations are always carried out before any operands are compared. The "while" loop in C is essentially the same syntactically as the WHILE-WEND structure in BASIC.

Indeed, C has some superior features that in many respects make it much more desirable than BASIC.

WHY CONVERT?

In spite of all these similarities, the two languages nonetheless have many important differences. Indeed, C has some superior features that in many respects make it more desirable than BASIC and, thus, worthwhile to translate BASIC programs to it. First, programs written in compiled C execute much faster than those written in either interpreted or compiled BASIC.

The advantages of C are not confined to speed, however. C enables the user to perform many tasks that simply cannot be done in BASIC. It is not just that C is a structured language that allows calls by name to newly improvised functions. With C's advanced pointer capabilities, for example, pointers to functions can be used. Thus, when separate routines are needed for different data types, they can be included in the same function and selected by pointers passed to the function when it is called.

AREAS OF DIFFICULTY

In making the change from BASIC to C, the programmer will have to master five tasks in addition to the C syntax. These are:

1. The compiling and linking environment
2. Structured programming practice
3. The complex set of C data types
4. The powerful uses of pointers in C
5. Learning to write functions.

As in any compiled language, once the C source code has been prepared with an editor, the source file must be passed through a compiler, which generates an object file of machine code. A linker must then be used to create a loadable and executable .EXE file. Learning the option parameters and correct syntax acceptable to the compiler and linker is a skill developed through experience.

In BASIC, the GOTO and GOSUB statements allow the user, in theory, to branch to any line number or subroutine anywhere in the program. He will, however, pay a considerable penalty in program speed and efficiency for lavish use of these unstructured branching functions. Structured programming involves careful design of the various segments of a program, according to the rules of the language, so that greater efficiency is possible. Although C offers a "goto" function, experienced C programmers seldom use it. Learning how to dispense with GOTOs and GOSUBs is one hurdle a BASIC programmer must jump when he is migrating from BASIC to C.

The C language is known for its large set of data types. Although some of these are much the same as the corresponding BASIC data types, some are entirely different and must be learned. C and BASIC have three data types in common: 2-byte integer variables; 4-byte single-precision floating-point variables; and 8-byte double-precision variables. In addition, C has seven other data types: auto, char, extern, long, short, register, and static. A variable must always be declared with its designated data type. If it is declared in the C preprocessor, it is global to the whole program. If it is declared in a function definition, then it is local to that function. For more information on the unique C data

types, consult one of the books recommended below.

With C, pointers come into focus as a programming technique that allows greater generality and compactness. A pointer is really nothing more than a variable that can hold the address of another variable. This additional dimension of variability gives another dimension of power. By using routines that manipulate the values of pointer variables, alternate variables can be passed to functions. Pointers can also be used as pointers to array elements. In general, anything that arrays can do, pointers can do faster. Also, arrays of pointers can be constructed and have many uses, such as in rapid sorting routines. Pointers play only a minor role in BASIC, but C is a language that almost appears to have been invented for the purpose of compact and powerful pointer manipulation.

Perhaps the most dramatic difference between C and BASIC derives from the fact that in C the programmer can write his own routines that then can be called by name as if they were new words in the language. Unlike a GOSUB routine, this technique allows the programmer to write functions that can

Pointers play only a minor role in BASIC, but C is a language that almost appears to have been invented for the primary purpose of compact and powerful pointer manipulation.

be compiled, placed in a library, and called by any C program. Writing these functions is a skill that must be acquired from experience. Thus, a BASIC programmer is unlikely to become proficient in C overnight.

Two software products now on the market are meant to make this

Table 1: Equivalent Functions of BASIC and BASIC_C

BASIC	BASIC_C	BASIC	BASIC_C
ABS	ABS	LPRINT	LPRINT
ASC	ASC	LPRINT USING	LPRINT USING
BEEP	BEEP	LSET	LSET
BLOAD	BLOAD	MID$ fcn	SUB_STR
BSAVE	BSAVE	MID$ stmt	MID
CHR$	CHR	MKD$	\
		MKI$	MK?
CLOSE	CLOSE	MKS$	/
	CLOSE_ALL	OCT$	OCT
CLS	CLS	ON ERROR	ON_ERR_GOTO
COLOR	COLOR		ON_ERR_HALT
CSRLIN	CSRLIN		ON_ERR_PRINT
CVI	\	OPEN	OPEN
CVS	CV?	OUT	OUT
CVD	/	PEEK	PEEK
DATE$ var	GET_DATE	POINT	POINT
DATE$ stmt	SET_DATE	POKE	POKE
DEF SEG	DEF_SEG	POS	POS
END	END	PRINT	PRINT
EOF	AT_EOF	PRINT USING	PRINT USING
ERR	ERR	PRINT#	FPRINT
ERROR	ERROR	PSET	PSET
FIELD	FIELD	PUT	PUT
GET	GET		PUT_NEXT
	GET_NEXT	RIGHT$	RIGHT
HEX$	HEX	SCREEN fcn	READ_SCREEN
INKEY$	INKEY	SCREEN stmt	SCREEN
	GETKEY	SGN	SGN
INP	INP	SPACE$	SPACE
INPUT	INPUT	SPC	in PRINT
INPUT#	FINPUT	STOP	STOP
INSTR	INSTR	STR$	NUM_STR
KILL	KILL	STRING$	STRING
LEFT$	LEFT	SWAP	SWAP
LEN	LEN	TAB	in PRINT
LINE INPUT	LINE_INPUT	TIME$ var	GET_TIME
LINE INPUT#	FLINE_INPUT	TIME$ fcn	SET_TIME
LOC	LOC	VAL	VAL?
LOCATE	LOCATE		STR_NUM
	CRSR_DEF	VARPTR	VARPTR
	CRSR_OFF	WRITE	BWRITE
	CRSR_ON	WRITE#	FBWRITE

Reproduced with permission from C Source Inc. © 1984.

Note:
fcn = BASIC function
stmt = BASIC statement
var = BASIC variable

transition easier: The BASIC_C Library and BASTOC.

THE BASIC_C LIBRARY

C Source Inc. has developed a product intended specifically for programmers with a working knowledge of BASIC who want to develop programming ability with the C language. It is a library for use with several C compilers on the IBM PC (currently Lattice, DeSmet, and Computer Innovations are supported) that implements many of the familiar BASIC statements and structures in the C environment. The purpose of this library is to ease the process of learning C for BASIC programmers, and to aid in the conversion of BASIC programs to C. It will not, however, enable a BASIC programmer to produce C programs without any knowledge of the C language. The BASIC_C Library manual is written with the C novice in mind and is one of the better introductions to C available, particularly for those familiar with the details of BASIC.

Version 1.0 of BASIC_C has about 75 functions in the library. Table 1 lists the corresponding BASIC and BASIC_C functions. No math functions are included. The

reason for this, according to a brochure issued by C Source, is that math functions belong in a standard library, "and only the Lattice/Microsoft C compiler does not have them." Version 2.12 of the Lattice compiler, which includes math functions, is now available.

For writing programs that do not require trig, log, exponential, or random functions, however, the BASIC_C Library is a welcome addition to the C environment for both the experienced C programmer and the novice. A loan payment calculation program serves as an example of a BASIC program (listing 1) translated to C (listing 2) using the BASIC_C Library. The example illustrates the conversion of a GOSUB into externally defined functions. It does not use any arrays, pointers, or the COLOR function.

Generally speaking, documentation for BASIC_C helps a BASIC programmer who has studied C syntax translate programs into C. Debugging these programs, however, is another matter. A programmer accustomed to BASIC will find that a new set of debugging skills is needed in the C environment. Also, C programs that are written in the style of the examples, virtual line-by-line translations from BASIC, are valuable as exercises in making the change from BASIC, but generally are not examples of good programming practices in C. For this reason, the gain in efficiency and speed over compiled BASIC may not be as great as anticipated. The good performance of C is due in part to its concise syntax. Using C to mimic BASIC does not take advantage of the full powers of C.

Using C to mimic BASIC does not take advantage of the full powers of C.

Although BASIC_C does an admirable job of getting a BASIC programmer started in making the transition to C, the product would be greatly improved if the documentation gave the user a better picture of what the final destination is. One way of doing this would be to provide more advanced examples that illustrate how the library can be used to write professional C programs, explaining the steps taken beyond the BASIC format of the simpler examples.

One relatively minor bug with the COLOR function could be a problem. Unlike the other functions in BASIC_C, COLOR does not behave the way its BASIC counterpart does. In IBM BASIC the COLOR statement can be used to set the color attributes of foreground, background, and border, clear the screen with CLS, and exit to DOS; the color attributes will remain in effect for the DOS environment. With the BASIC_C COLOR func-

How Can You Avoid Getting Trapped Under An Ancient Word Processor?

Buy Word Perfect!

Once upon a time, word processors were monstrous things. Dot commands, page orientation, and separate editing, formatting and printing programs turned them into lumbering beasts. Only a well-educated programmer would dare don his armor and tackle such a beast — not a pleasant task for a modern secretary, executive, or writer.

Then came WordPerfect and the beast was slain.

WordPerfect was designed to work for you not against you. WordPerfect has no command language to complicate your writing. Pressing a single key is all it takes to

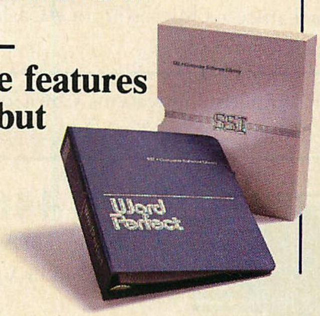
bold, underline or center.

When writing, you don't want to worry about page formatting, making room for headers or footers, or whether you are in "edit" or "create" mode. Your word processor should do it automatically and WordPerfect does. WordPerfect lets you think in terms of ideas, not pages. It is simple enough that you quickly forget about the mechanics and your writing flows easily.

So if you don't want to be caught under a word processing monster, try WordPerfect. We're certain it will improve the quality of your writing.

WordPerfect

You'll love it — not only for the features we've built in, but also for the antiquities we've left out.



SATELLITE SOFTWARE INTERNATIONAL

288 WEST CENTER STREET, OREM, UTAH 84057 (801) 224-4000 TELEX 820-618

SSI

CIRCLE NO. 236 ON READER SERVICE CARD

tion, on the other hand, a print statement has to be issued for the color attributes to appear, and it is not possible to set the border color and keep it after exiting to DOS. These are not very serious problems, but they should either be fixed or be properly documented.

For experienced C programmers, the BASIC_C Library has some handy functions that are rarely found for the IBM PC. Among them are BLOAD(), BSAVE(), SET_TIME(), GET_TIME(), SWAP(), POINT(), and BEEP. Many of these functions, for example, are not included in the Greenleaf Functions, by Greenleaf Software, Inc., which claims to be the most inclusive C library.

BASTOC

JMI Software Consultants have ported to the IBM PC a product that produces C source code from BASIC ASCII files (minus interpreter-oriented functions and those that interface to MS-DOS). Generic MS-BASIC falls short of PC BASIC, but many programs written in it can run under the PC's BASIC interpreter. The main intended PC-related use of the translator is to help move programs first written in BASIC in the CP/M environment into C on the PC.

Three versions are available for the IBM PC: the Computer Innovations version 1.33D, the Lattice 2.0, and Quantum 1.1 C compilers. The version evaluated here is BASTOC 1.4 for the Lattice 2.0 compiler. JMI offers a source license for BASTOC at a price of $15,000, and maintenance is set at $3,000 per year.

The BASTOC system has two parts: the translator program and its accompanying library. There are three main modes or options available in using BASTOC. It can produce a monolithic program with all code imbedded in one main segment, it can produce a structured program in which the GOSUBs are decomposed into a number of functions, or it can convert the BASIC program into a C function. The user also may customize the BASTOC system in various ways.

BASTOC comes on three double-sided floppy diskettes and includes a rather short, spiral-bound booklet for documentation. The files provided on the diskettes include the translator program and both translator and link libraries for all four of the Lattice 2.0 memory models. Also included is the object module for the translator, several header files, and the C source for syntax and parameter tables, so that BASTOC can be modified to suit different purposes. The translator is written in and produces machine-independent C that adheres to the standard specified in *The C Pro-*

Table 2: BASIC Language Features Supported by BASTOC

STATEMENTS

CALL ident [([argument] {, argument})]
CHAIN file name
CLEAR [, string space [, stack space]]
CLOSE [[#] number {, [#] number}]
CLS
DATA datum {, datum}
DEF FNx (id {, id}) = expression
DEF FNx (id {, id})
DEFDBL letters {, letters}
DEFINT letters {, letters}
DEFSNG letters {, letters}
DEFSTR letters {, letters}
DEFUSR[n] = number
DIM array-decl {, array-decl}
END
ERASE array name {, array name}
ERROR number
FEND
FIELD [#] number, fwidth AS svar {, fwidth AS svar}
FILES [pattern]
FOR v = initial-value TO limit [STEP increment]
GET [#] number, recno
GOSUB line number
GOTO line number
If expression THEN statements [ELSE statements]
INPUT [string-constant;] variable {, variable}
INPUT # number, variable {, variable}
INTEGER identifier-list
KILL file name
[LET] variable = expression
LINE INPUT [string-constant;] variable {, variable}
LINE INPUT # number, variable {, variable}
LPRINT [USING format] . . .
LSET name = string
MID$(old string, position [, length]) = new string
NAME old file AS new file
NEXT [variable {, variable}]
NEW
ON ERROR GOTO line number
ON expression GOSUB line number {, line number}
ON expression GOTO line number {, line number}
OPEN "I", [#] number, string
OPEN "O", [#] number, string
OPEN "R", [#] number, string
OPTION BASE
OUT port number, data-byte
POKE addr, value
PRINT [# number,] [USING format,] expr {p expr} [p]
PRINT@ number, . . .
PRINT@ (number, number), . . .
PUT [#] number, recno
RANDOMIZE [expression]
READ variable {, variable}
REM . . .
RESTORE
RESUME [line number] and RESUME NEXT
RETURN
RSET name = string
RUN
STOP
SWAP variable1, variable2
TROFF
TRON
WAIT port number, mask [, pattern]
WEND
WHILE expression
WIDTH [filenum,] size and WIDTH device name, size
WRITE [#filenum,] expr {, expr}

OPERATORS

+	add
*	multiply
^	exponentiate
=	equals
<=	less than or equal
<	less than
NOT	bitwise negation
OR	bitwise disjunction
IMP	bitwise implication

gramming Language (Kernighan and Ritchie, 1978).

The BASIC statements and functions supported in the current release of BASTOC are listed in table 2. Particularly important for those wanting to translate IBM PC BASIC programs is the absence of the SCREEN, COLOR, LOCATE, KEY, COM, PRINT USING, and BEEP statements. The translator does recognize the PRINT@ construction for printing at a specified screen location, so it is possible either to edit existing BASIC programs that use the LOCATE statement or to write new programs targeted at the translator.

The translator program itself offers a total of seven different options for its output:

1. To make blanks significant as separators between key words.
2. To insert the BASIC source as comments in the C code
3. To translate the BASIC code into one or more functions
4. To produce one monolithic C function instead of mapping each GOSUB into a separate function
5. To use a specified file name as the output file
6. To convert BASIC integers to C variables of the "short" data type
7. To issue statement trace directives (statements only).

The best features of the BASTOC translator are its speed, its variety of options, and its capability to produce a C source file no matter what, with the BASIC lines it was unable to translate inserted in the code as comments. Especially impressive is its speed. Unless there are a large number of error messages, the translator will generally finish a given program considerably sooner than would Lattice 2.0. The error messages specify the problem and print the troublesome BASIC line with a dotted line pointing, if necessary, to the problem location.

The resultant C code is well formatted, but unfortunately the label and function names are unfamiliar and not easy to read, which severely limits the usefulness of the generated code. An example of a BASIC program that translated as is into C, and that compiled, linked, and executed successfully, is provided in listing 3 (BASIC) and listing 4 (C).

MOD integer remainder
− subtract
/ divide
\ integer divide
<> not equal
>= greater than or equal
> greater than
AND bitwise conjunction
XOR bitwise exclusive OR
EQV bitwise equivalence

FUNCTIONS

ABS(X#)—absolute value
ASC(S$)—ASCII code for first character of S$
ATN(X#)—arc tangent
CDBL(expression)—converts expression to double precision
CHR$(N%)—character with ASCII code N%
CINT(expression)—converts expression to integer
COS(X#)—cosine (of an angle in radians)
CSNG(expression)—converts expression to single precision
CVD(F$)—convert F$ to double precison
CVI(F$)—convert F$ to integer
CVS(F$)—convert F$ to single precision
DATE$—current date
EOF(N%)—indicate end of file condition
ERL—line number where last error occurred
ERR—error code of last error that occurred
EXP(X#)—exponential function
FIX(X#)—integer part of X#
FRE—available free storage
HEX$(N%)—convert N% to hexadecimal string
INKEY$—read a character from the standard input
INP(N%)—input a data byte from port N%
INPUT$(N%, [#] filenum])—input N% bytes from a file or keyboard
INSTR([N%,] S$, P$)—position of P$ in S$
INT(X#)—largest integer less than X#
LEFT$(S$, N%)—leftmost N% characters of S$
LEN(S$)—number of characters in S$
LOC(filenum)—returns record number of last processed record in file
LOF(N%)—length of file in records
LOG(X#)—natural logarithm
LPOS(N%)—position of line printer print head
MID$(S$, I%, L%)—L% characters of S$ starting with I%
MKD$(N#)—convert double precision to string
MKI$(N%)—convert integer to string
MKS$(N!)—convert single precision to string
OCT$(N%)—convert N% to octal string
PEEK(N%)—reads byte in memory location N%
POS—position of cursor
RIGHT$(S$, N%)—rightmost %N characters of S$
RND()—random number
SGN(X#)—algebraic sign of X#
SIN(X#)—sine (of an angle in radians)
SPACE$(N%)—string of N% spaces
SPC(N%)—print N% spaces in a PRINT or LPRINT statement
SQR(X#)—square root
STR$(N#)—returns string expression of N#
STRING$(N%, M%)—string of N% ASCII M% characters
STRING$(N%, S$)—string of N% S$ characters.
TAB(N%)—tabs to column N% in a PRINT or LPRINT statement
TAN(X#)—tangent (of an angle in radians)
TIME$—current time
USR[n] [argument]—call a user function
VAL(S$)—returns numeric value of S$
VARPTR(variable)—address of variable in memory

COMPILER DIRECTIVES

%INCLUDE name—include named source file
%LIST, %NOLIST—allowed for compatibility with CBASIC, but ignored

The BASTOC translator has a number of limitations and idiosyncracies—some of which are documented but many of which are not. An important documented limitation is that the translator will not handle nested subroutines or multiple entry points within a given subroutine. In either case, an error message is sent to the screen during translation and imbedded in the C source as a comment.

Another undocumented limitation is that BASTOC does not recognize various syntactic nuances that BASIC programmers have devised for optimizing their code, such as using NEXT instead of, say, NEXT I. Neither does BASTOC recognize the use of the % symbol for declaring integer variable values. An apparent bug prevents BASTOC from handling the READ . . . DATA construction properly. Programs that used this construction translated, compiled, and linked with no error messages. At run time, however, the executing program stopped to deliver a message that it was not given sufficient data, even though all of the BASIC data statements were present in corresponding C statements in the source file.

The best features of the BASTOC translator are its speed, its variety of options, and its ability to produce a C source file no matter what the difficulties are.

Generally, BASTOC should not be expected to translate many programs flawlessly so that they will compile and execute, without some meticulous editing of the BASIC or C source code. Of more than 20 programs scrupulously written in Micosoft BASIC dialect, only five produced executable C files that performed as the BASIC programs did. Two of the five were examples provided by JMI. Once, inexplicably, BASTOC failed to include a required C routine; this omission made the resulting program function improperly, even though no error messages were given during translating, compiling, or linking.

In spite of these limitations, BASTOC can be a valuable tool for an experienced C programmer. For large programs, probably the only workable procedure is to choose the option that breaks the BASIC code into a number of separate C functions. In this case, the programmer will have a number of usable modules right away. The faulty modules can then be isolated, edited in the BASIC or C source, or rewritten using BASIC_C or one of the other available C libraries. Finally, a main segment can be added that calls all the separate function modules. There should be few BASIC programs that cannot be moved to C in this way. The user must decide whether this method is preferable to rewriting a program.

The documentation for the BASTOC package gives examples of how to add statements, functions, and operators to the system. Several steps are involved. First, the new function must be written in C and compiled. An entry must be added to the appropriate key-word array in the provided file ATLBLS1.C and compiled. The BASTOC manual fails to mention that if any new declarations of parameters are needed, these have to be added to the header file AT.H. Otherwise, the file will not compile.

A librarian utility (see "Checking Out the Librarians," Kuryan Thomas, p. 80 in this issue) is needed to add the object file for the newly created function to the BASTOC link library. Finally, the new modules must be linked into a newly created translator program. In a test this customizing capability was able to add the MAX function to BASTOC, but the translator refused to recognize the CLRBOX statement, even though all the steps were carried out.

The possibility that existing object files emulating BASIC functions, such as those in the BASIC_C Library, can be added to BASTOC is a tantalizing one. To try it, the

Polylibrarian utility was used to extract the object files for several statements from the BASIC_C Library and add them to the BASTOC link library. The attempt, unfortunately, was not successful. Just as in the case of the CLRBOX statement, BASTOC refused to recognize these new statements.

Programmers inexperienced with C will likely have trouble customizing BASTOC. In fact, even professional software engineers who are well-versed in C may discover that their efforts are frustrated.

Programmers inexperienced with C will likely have trouble customizing BASTOC. In fact, even professional software engineers who are well-versed in C may find their efforts frustrated.

BASIC TO C SYNERGY

Although the attempt to add functions from the BASIC_C Library to BASTOC did not succeed, these two products can complement one another in a number of ways. This is an area now being explored.

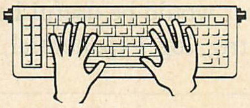
As mentioned earlier, if a modular approach is used for moving large programs from BASIC into C by using the BASTOC option that maps BASIC modules into separate C functions, then BASIC_C can be used to recode those modules that don't translate successfully. On the other hand, the BASTOC libraries can also be used to complement the statements supported in BASIC_C. This would be done by calling functions in the BASTOC library as well as BASIC_C in C programs or functions that the user writes.

The unfamiliar function calls and the lack of documentation about them prevents this from be-

Get Your Hands On Our Hands-On Analysis

Introducing Data Decisions™ MICROCOMPUTERS

An **ALL-NEW** information service for the corporate micro buyer that features



Hands-On-Testing

of Micro Software

by Experts in Data Decisions™ OWN LABS

Ask for a trial review of MICROCOMPUTERS and let Data Decisions do your software testing for you. You'll get the results of hundreds of hands-on tests **done under consistent, controlled conditions** by our team of micro experts . . . You'll find out how competing packages measure up against each other and what you can expect when you bring them in-house.

And that's just the beginning! Along with everything you need to know about spreadsheets . . . graphics . . . word processors . . . and other software products—you also get the latest word on business microcomputer systems . . . expansion/extension cards . . . communications products . . . peripherals—all the products and technologies you need to know about to grow.

MICROCOMPUTERS comes to you in three monthly updated loose-leaf volumes that are **consistently formatted** for easy product comparisons . . . and for finding the facts you need FAST.

Whether you're looking at LANs, PC compatibles, printers for micros, or the new integrated software packages, Data Decisions MICROCOMPUTERS supports you every step of the way. We have:

- **Hands-on evaluations** of software products by micro experts.
- **Rating charts** that let you compare how competing packages measure up in actual use.
- **In-depth analyses** that highlight strengths and limitations of mainstream products.
- All the **pricing** and **spec's** you need for feasibility studies and budgets.
- **Technology reports** that fill you in on issues and trends.
- **Unlimited free telephone consulting** by micro experts whose only job is to answer your questions.
- **Broadbased surveys . . . monthly newsletters . . . complete vendor directory . . . MORE.**

A 1-year subscription to Data Decisions MICROCOMPUTERS costs only $700. You get all this information and analysis for less than it would cost you to hire a consultant to test just one product!

But don't take our word for it . . .

SEE FOR YOURSELF!

Examine MICROCOMPUTERS at your leisure in your own office before you buy. Just fill in the coupon and mail it today. You'll receive a no-purchase-obligation TRIAL REVIEW.

FREE BONUS REPORT: And as an added bonus . . . just for taking a trial review of MICROCOMPUTERS, we'll also send you a free copy of

Business Communication Networks—
Putting It All Together

This special information-packed technology report shows you how to build a user network—from policy planning . . . to the selection and integration of products . . . to network management and control. It regularly sells for $39.00. . . . But it's yours to keep ABSOLUTELY FREE—Just for agreeing to review MICROCOMPUTERS . . . So fill in the coupon TODAY!

CIRCLE NO. 126 ON READER SERVICE CARD

Data Decisions™ 20 Brace Road/Cherry Hill, NJ 08034/609-429-7100

We do the hands-on testing for you

☐ C5 Yes, please send me my TRIAL REVIEW of MICROCOMPUTERS. I understand that I'm under no obligation to purchase anything and that I may keep the free report even if I don't become a subscriber.

☐ Y4 Although your offer is completely risk-free, I'm still not sure. Please provide more information.

Data Decisions™ 20 Brace Road/Cherry Hill, NJ 08034/609-429-7100

Name____________________

Title____________________

Company____________________

Address____________________

City__________ State______ Zip______

Phone (________) ________________
Area Code

A107

ing as straightforward as it should be, but with some study of the BASTOC output code it should not be too difficult. A header file that mapped the obscure names of the BASTOC functions into more familiar ones would make the process easier. Fortunately, many of the functions that would be useful here, such as the trig, log, exponential, and random math functions, already have familiar names.

Another way that BASTOC can help write programs with BASIC_C is by providing examples of various strategies for translating a given program. Although the program or function may not have been automatically translated without error, BASTOC usually provides C code that is close to what is needed. These listings can be quite valuable as references in planning a conversion strategy, and in some cases BASIC_C may be used to revise the BASTOC C code rather than completely rewriting it.

Although the process of translating programs from BASIC to C is hardly a matter of a quick pass through a code generator, BASIC_C and BASTOC are two welcome additions in the C environment that make it far easier than before to move programs between these two very popular programming languages.

BASIC_C
C Source Inc.
12801 Frost Road
Kansas City, MO 64138
816-353-8808
$175 (including source code)
CIRCLE 494 ON READER SERVICE CARD

BASTOC
JMI Software Consultants Inc.
P.O. Box 481
Spring House, PA 19477
215-628-0840
$350
Source code: $15,000
CIRCLE 495 ON READER SERVICE CARD

Recommended Books

Kernighan, B., and D. Ritchie, *The C Programming Language*, Englewood Cliffs: Prentice-Hall, 1978.

Plum, Thomas, *Learning to Program in C*, Cadiff, NJ: Plum Hall, 1983.

Purdum, Jack, *C Programming Guide*, Indianapolis: Que Corporation, 1983

Purdum, Jack, *C Programmers Library*, Indianapolis: Que Corporation, 1984.

THE WORD FILE CONNECTION

8" DISKETTE SYSTEM FOR THE IBM PC

8" Diskette file exchange between the IBM PC and most micro-mini-mainframe computer systems.

8" Diskette text document exchange between the IBM PC and many word processing systems.

8" Diskette to 5¼" formats for hundreds of other systems, even Textpack and Wordstar to the new DisplayWrite 2 format.

FLAGSTAFF ENGINEERING / P. O. Box 1970 / Flagstaff, AZ 86002
Telephone 602-774-5188 / Telex 705609 FLAG-END-UD

CIRCLE NO. 213 ON READER SERVICE CARD

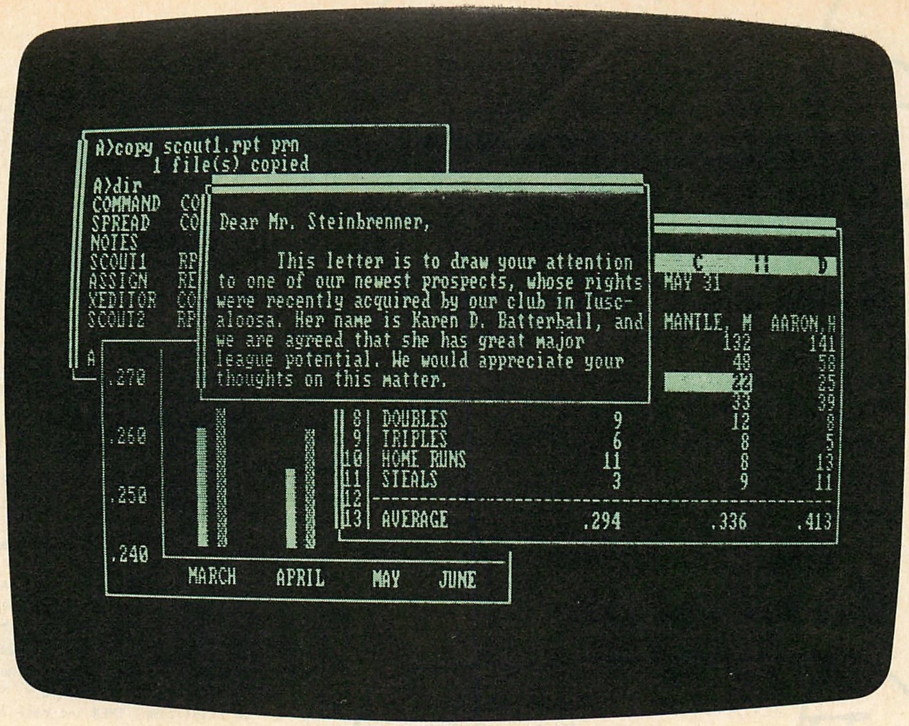
Listing 1 BASIC Source for Loan Payment Calculation Program

```
100 GOSUB 800
110 REM
120 REM     ********** PRINTS COLUMN LINES **********
130 REM
140 FOR T=1 TO 18
150 PRINT TAB(5);
160 GOSUB 860
170 NEXT T
180 REM
190 REM
200 ANSWER$ = "YES"
210 WHILE ANSWER$ = "YES" OR ANSWER$ = "yes"
220       LOCATE 24,5
230 REM     ********** PROMPTS USER FOR INFORMATION **********
240       INPUT; "PRINCIPAL -  ",TEMP
250       IF TEMP<>0 THEN ORG.PRNCPL=TEMP
260       INPUT; "     NO. OF MONTHS -  ",TEMPOR
270       IF TEMPOR<>0 AND TEMPOR<19 THEN MNTHS=TEMPOR ELSE GOTO 210
280       INPUT; "     INTEREST RATE -  ",TEMP
290       IF TEMP<>0 THEN INT.RATE=TEMP
300       CLS
310       LOCATE 1,1
320       GOSUB 800
330 REM
340 REM     ********** CALCULATES VALUES FOR COLUMNS **********
350 REM
360       MINT.RATE = INT.RATE/1200
370       CUM.INT = 0
380       WT.FCTR = (1-(1/(1+MINT.RATE)^MNTHS))
390       MNTH.PYMT = MINT.RATE*ORG.PRNCPL/WT.FCTR
400       PRNCPL = ORG.PRNCPL
410 REM
420 REM     ********** CALCULATES VALUES FOR EACH MONTH **********
430 REM
440       FOR N=1 TO MNTHS
450             INT.PYMT = PRNCPL*MINT.RATE
460             PRNCPL.PYMT = MNTH.PYMT-INT.PYMT
470             CUM.INT=CUM.INT+INT.PYMT
480             PRINT TAB(7);
490             PRINT USING "##     |";N;
500             PRINT TAB(13);
510             PRINT USING "#####,.##    |";PRNCPL;CUM.INT;INT.PYMT;
520             PRINT USING "#####,.##";PRNCPL.PYMT;MNTH.PYMT
530             PRNCPL = PRNCPL-PRNCPL.PYMT
540             NEXT N
550       LOCATE 23,5
560       INPUT "DO YOU WISH TO MAKE ANY CHANGES"; ANSWER$
570 WEND
580 END
590 REM
600 REM     ********** PRINTS THE HEADER **********
610 REM
800 PRINT
810 PRINT TAB(12) "|" TAB(25) "| CUMULATIVE |  INTEREST     ";
815 PRINT "PRINCIPAL      MONTHLY"
820 PRINT TAB(4) "MONTH    | PRINCIPAL |  INTEREST  |  PAYMENT";
825 PRINT "   +    PAYMENT   =   PAYMENT"
830 X$ = STRING$(72,95)
840 PRINT TAB(4);X$:PRINT TAB(5)
850 RETURN
860 PRINT TAB(12) "|" TAB(25) "|" TAB(38) "|" TAB(51) "|" TAB(64) "|"
870 RETURN
```

Reproduced with permission from C Source Inc. ©1984.

Listing 2 BASIC_C Output for Loan Payment Calculation Program

```
#include <bas.h>

main()
{

INT t;

/*        ********** prints column lines **********          */
INITBC();
```



SOFTWARE FUSION

APX

CORE EXECUTIVE™

INTEGRATE VIRTUALLY ANY SOFTWARE

The APX CORE EXECUTIVE enables you to:

★ Run several programs simultaneously.

★ Control each program through interactive windows.

★ Change any window's size and position, and scroll within any window.

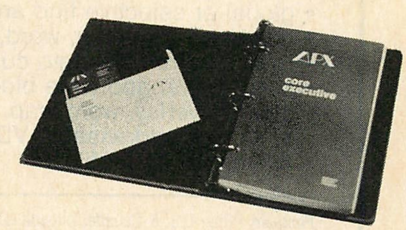
★ "Cut & paste" from one window to another automatically.

★ Save typed sequences for replay with a single keystroke.

All this, and more.... **$195.00**

The APX CORE EXECUTIVE (available for the IBM PC with DOS).

Dealer inquiries invited.



Application Executive Corporation

600 Broadway
Suite 4C
New York, NY 10012
(212) 226-6347

APX

CIRCLE NO. 192 ON READER SERVICE CARD

```
header();
for (t = 0; t < 18; t++){
        PRINT("T;", 5);
        prt_column();
        }
calc_pymt();
}


calc_pymt()
{
INT mnths, n, tempor;
DBL temp, org_prncpl, int_rate, mint_rate, cum_int, power();
DBL prncpl, int_pymt, prncpl_pymt, wt_fctr, mnth_pymt;
STR answer;

answer = "YES";
while (CMP(answer, IS_EQ, "YES") OR CMP(answer, IS_EQ, "yes")){
    LOCATE(24, 5);
/*
 *        **********  prompts user for information  **********
 *                                                             */
    INPUT("PRINCIPAL -  ", ";D", &temp);
    IF ((INT)temp) THEN org_prncpl = temp;
    INPUT("     NO. OF MONTHS -  ", ";I", &tempor);
    IF (tempor > 16) THEN continue;
    IF (tempor) THEN  mnths = tempor;
    INPUT("     INTEREST RATE -  ", ";D", &temp);
    IF ((INT)temp) THEN int_rate = temp;
    CLS();
    LOCATE(1,1);
    header();
    prt_column();
/*
 *        **********  calculates values for columns  **********
 *                                                             */
    mint_rate = int_rate/1200;
    cum_int = 0;
    wt_fctr = (1-(1/power((1 + mint_rate), mnths)));
    mnth_pymt = mint_rate * org_prncpl / wt_fctr;
    prncpl = org_prncpl;
/*
 *      **********  calculates values for each month  **********
 *                                                             */
    for (n = 1; n <= mnths; n++){
    int_pymt = prncpl * mint_rate;
    prncpl_pymt = mnth_pymt - int_pymt;
    cum_int = cum_int + int_pymt;
    PRINT("T;", 7);
    PRINT_USING("--   _|;", 1, n);
    PRINT("T;", 13);
    PRINT_USING("#####,.##   _|;", 4, prncpl, cum_int, int_pymt,
                             prncpl_pymt);
    PRINT_USING("#####,.##;", 1, mnth_pymt);
    prncpl = prncpl - prncpl_pymt;
    }
  LOCATE(23, 5);
  INPUT("DO YOU WISH TO MAKE ANY CHANGES?  ", "S", &answer);
  }
```

Reproduced with permission from C Source Inc. ©1984.

Listing 3 BASIC Source for Inventory Program

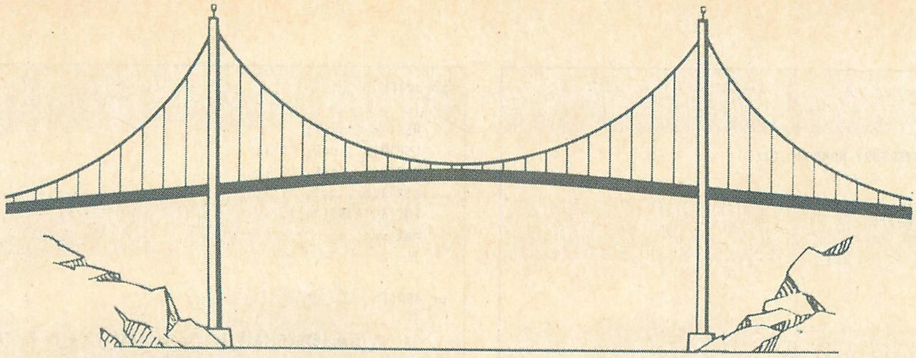
```
10 OPEN "R", 1, "INVENTORY", 34
20 PRINT "MENU"
30 PRINT "1: ADD A NEW ITEM"
40 PRINT "2: RETRIEVE AN ITEM"
50 PRINT "3: CHANGE AN ITEM"
60 PRINT "4: DELETE AN ITEM"
70 PRINT "5: FINISH"
80 INPUT "YOUR CHOICE";X
90 ON X GOSUB 110,140,170,220,270
100 GOTO 20
110 GOSUB 290
120 GOSUB 330 : GOSUB 390
130 RETURN
140 GOSUB 290
150 GOSUB 450
```

```
160 RETURN
170 GOSUB 140
180 PRINT "NEW DATA"
190 GOSUB 330
200 GOSUB 390
210 RETURN
220 GOSUB 140
230 DE$ = "DELETED"
240 INPUT "OK TO DELETE (Y/N)";AN$
250 IF AN$ = "Y" THEN GOSUB 390
260 RETURN
270 CLOSE
280 END
290 INPUT "INVENTORY NUMBER";IN
300 FIELD 1,20 AS A$,8 AS B$,4 AS C$,2 AS D$
310 GET 1,IN
320 RETURN
330 LINE INPUT "DESCRIPTION? ";DE$
340 DE$ = LEFT$(DE$, 20)
350 INPUT "QUANTITY";QU%
360 INPUT "PRICE";PR#
370 INPUT "COST";CO!
380 RETURN
390 RSET A$=DE$
400 LSET B$=MKD$(PR#)
410 LSET C$=MKS$(CO!)
420 LSET D$=MKI$(QU%)
430 PUT 1, IN
440 RETURN
450 IF RIGHT$(A$,7) = "DELETED" THEN PRINT "DELETED" : RETURN
460 PRINT "DESCRIPTION";TAB(22);"PRICE";TAB(30);"COST";TAB(38);"QUANTITY"
470 PRINT A$;TAB(22);CVD(B$);TAB(30);CVS(C$);TAB(38);CVI(D$)
480 RETURN
```

Listing 4 BASTOC Output for Inventory Program

```
char *data_stmts[] =
    {
    0
    };
extern double brd_num();
extern double CVD();
extern CVI();
extern double CVS();
extern char *FIELD();
extern GET();
extern char *LEFT_();
extern PUT();
extern char *RIGHT_();
extern char *SUBFLD();
extern TAB();
extern char *brd_line();
extern char *brd_str();
extern char *fld_get();
extern char *s_asgn();
char * a_;
char * an_;
char * b_;
char * c_;
float coF;
char * d_;
char * de_;
float in;
double prD;
int quI;
float x;
static char nul_str[1] = {0};
static char *st_1 = nul_str, *st_2 = nul_str;
static char *fl_1 = 0;
main(argc, argv)
    int argc;
    char *argv[];
    {
    bio_init(argc, argv);
    bassoc("INVENTORY",1);
    bda_fset(1, 34);
  l_20: ;
    bwr_set(1);
```

```
bwr_str("MENU", 0);
bwr_nl();
bwr_set(1);
bwr_str("1: ADD A NEW ITEM", 0);
bwr_nl();
bwr_set(1);
bwr_str("2: RETRIEVE AN ITEM", 0);
bwr_nl();
bwr_set(1);
bwr_str("3: CHANGE AN ITEM", 0);
bwr_nl();
bwr_set(1);
bwr_str("4: DELETE AN ITEM", 0);
bwr_nl();
bwr_set(1);
bwr_str("5: FINISH", 0);
bwr_nl();
do      {
        bwr_prompt("YOUR CHOICE",0);
        brd_set(0, 1);
        x = brd_num();
        } while(brd_err());
switch(CINT(x))
        {
  case 1: pr_110(); break;
  case 2: pr_140(); break;
  case 3: pr_170(); break;
  case 4: pr_220(); break;
  case 5: pr_270(); break;
        }
goto l_20;
}

pr_110()
  {
  pr_290();
  pr_330();
  pr_390();
  return;
  }
pr_140()
  {
  pr_290();
  pr_450();
  return;
  }

pr_170()
  {
  pr_140();
  bwr_set(1);
  bwr_str("NEW DATA", 0);
  bwr_nl();
  pr_330();
  pr_390();
  return;
  }

pr_220()
  {
  pr_140();
  de_ = s_asgn(de_, "DELETED");
  do      {
          bwr_prompt("OK TO DELETE (Y/N)",0);
          brd_set(0, 1);

          an_ = brd_str(an_);
          } while(brd_err());
  if(-(s_comp(an_,"Y")==0))
          {
          pr_390();
          }
  return;
  }

pr_270()
  {
  bclall();
  bexit(0);
  }
```



PELADA TEXT ENGINEER

The first full featured word processor for the IBM personal computer capable of producing all mathematical, scientific and engineering symbols* including superscripts and subscripts for complete equation editing on screen.

Multi-lingual capabilities include the specific characters, accents and letters found in French, Spanish, German, Italian and all other western European languages.

Advanced Word Processing features include:

- Command line driven editor
- Comprehensive, easily accessible on-line help
- All features, including bold, italics, underlining and special characters appear on screen exactly as they are printed.

SPECIAL INTRODUCTORY PRICE $250 MAIL ORDER ONLY. OFFER EXPIRES JULY 31/84.

PELADA INFORMATICA INC.

87 Queen St., Kingston, Ontario,
Canada K7K 1A5
Tel. (613) 549-1747

*Requires Epson FX-80, FX-100 or similarly featured printer.

FREE SAMPLE DISK AVAILABLE UPON REQUEST.

CIRCLE NO. 141 ON READER SERVICE CARD

```
pr_290()
   {
   do       {
            bwr_prompt("INVENTORY NUMBER",0);
            brd_set(0, 1);
            in = brd_num();
            } while(brd_err());
   fl_1 = FIELD(fl_1, 1,4);
   a_ = SUBFLD(20);
   b_ = SUBFLD(8);
   c_ = SUBFLD(4);
   d_ = SUBFLD(2);
   GET(1, CINT(in));
   return;
   }

pr_330()
   {
   bwr_prompt("DESCRIPTION? ",0);
   brd_set(0, 0);
   de_ = brd_line(de_);
   de_ = s_asgn(de_, LEFT_(&st_1, de_, 20));
   do       {
            bwr_prompt("QUANTITY",0);
            brd_set(0, 1);
            quI = brd_int();
            } while(brd_err());
   do       {
            bwr_prompt("PRICE",0);
            brd_set(0, 1);
            prD = brd_num();
            } while(brd_err());
   do       {
            bwr_prompt("COST",0);
            brd_set(0, 1);
            coF = brd_num();
            } while(brd_err());
   return;
   }
```

```
pr_390()
   {
   RSET(a_, de_);
   LSETD(b_, prD, 'D');
   LSETD(c_, coF, 'S');
   LSETI(d_, quI, 'I');
   PUT(1, CINT(in));
   return;
   }

pr_450()
   {
   if(-(s_comp(RIGHT_(&st_1, fld_get(&st_2, a_), 7),"DELETED")==0))
            {
            bwr_set(1);
            bwr_str("DELETED", 0);
            bwr_nl();
            return;
            }
   bwr_set(1);
   bwr_str("DESCRIPTION", 0);
   TAB(22);
   bwr_str("PRICE", 0);
   TAB(30);
   bwr_str("COST", 0);
   TAB(38);
   bwr_str("QUANTITY", 0);
   bwr_nl();
   bwr_set(1);
   bwr_str(fld_get(&st_1, a_), 0);
   TAB(22);
   bwr_num(CVD(b_), 0);
   TAB(30);
   bwr_num(CVS(c_), 0);
   TAB(38);
   bwr_int(CVI(d_), 0);
   bwr_nl();
   return;
   }
```


CIRCLE NO. 116 ON READER SERVICE CARD

NOVATION

TALKS BUSINESS

Introducing the LSI Cat Communication System™ with the Hayes compatible 300/1200 *Smart-Cat* **Plus**™ modem.

No more wondering what goes with what. Here's the best modem and the best software for your specific computer—all in one package.

Will your computer take an internal modem? Or do you prefer a stand-alone? And which operating system do you need—CP/M, CP/M-86, PC-DOS, MS-DOS? The Cat System includes them all.

And no compromises.

The quality starts with our new Smart-Cat PLUS modem. There is absolutely nothing else to equal it on the market. You get faster more accurate dialing. Automatic self-testing of the full communications circuit. State-of-the-art advanced LSI circuitry. And more. It is simply the best there is.

CIRCLE NO. 103 ON READER SERVICE CARD

The software is MITE™—menu-driven, easy-to-use, remarkably powerful. It gives you error-free data communications and instant access to Dow Jones, CompuServe and the like.

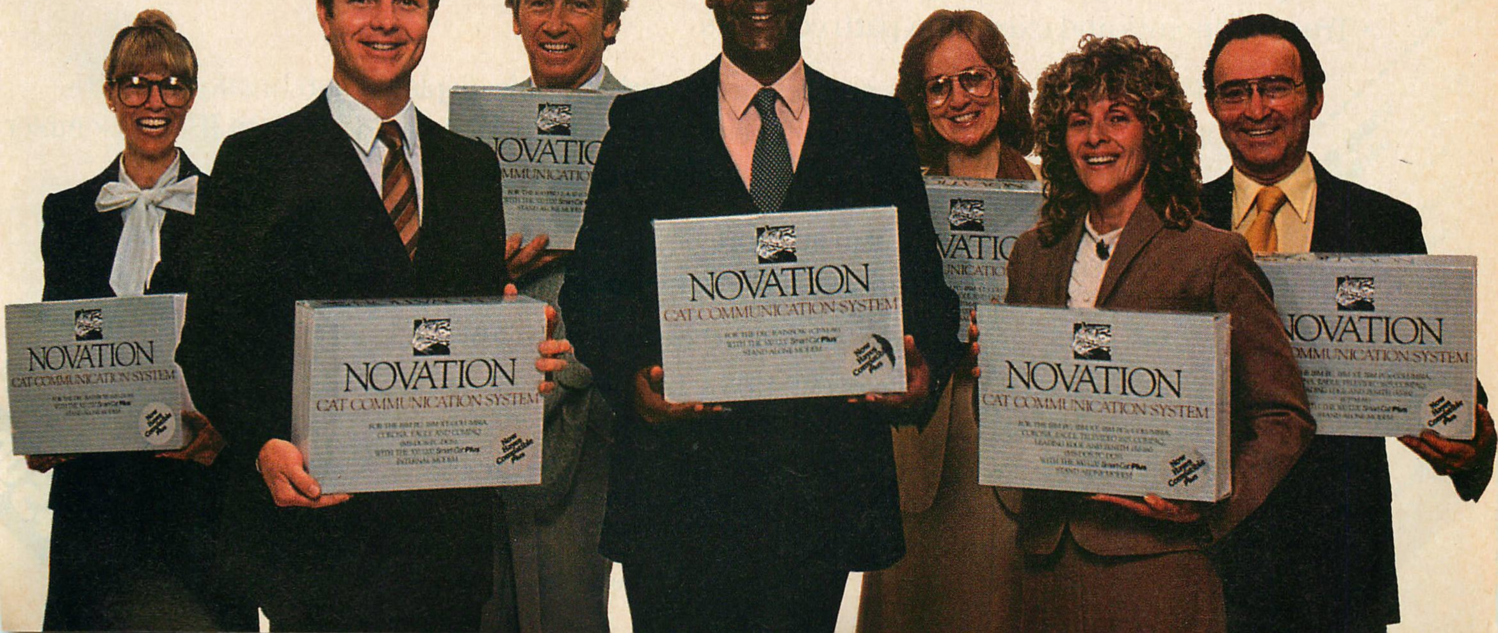
With the Cat System everything you get is the best of everything—that includes the best price. Suggested retail: $499.

See your dealer. He has them right now.

Seven models covering the IBM-PCs and their look-alikes, DEC Rainbow, KAYPRO and their operating systems. More models to come.

Novation, Inc., Box 2875
20409 Prairie Street, Chatsworth, CA 91311
(800) 423-5419 • In California: (818) 996-5060

NOVATION



XCOM

IBM PC MODEM

CROSSTALK XVI™

XCOM is an internally mounted, 300/1200 bps modem that includes CROSSTALK XVI communications software. Designed for use in the IBM PC and compatibles, XCOM comes with everything you need for your PC to start communicating with other computers: user's manual, CROSSTALK diskette, telephone cable, card guide and even a 2 year warranty.

XCOM automatically selects between tone and pulse dialing, checks for dial tone, detects ring-back signals and reports busy signals — an ease of use unheard of before. The simulated speaker function allows you to listen to your calls without watching your screen. Our switch-to-voice capability lets you use your PC for voice calls and for auto-dialing your frequently used numbers.

CROSSTALK XVI software, the industry standard, assures error-free file transfers with most microcomputer communication systems. A user-defined procedure, with a single command, configures the hardware, initiates the call, and logs in. With CROSSTALK, the IBM PC or compatibles can emulate some of the industry's more popular terminals.

Our price, $449, plus our unique features, makes XCOM the intelligent choice. Contact us for the dealer/distributor in your area.

ΔCOM

Delta Communication Products, Inc.
3213 Ramos Circle
Sacramento, CA 95827
Telephone: 916-366-7071

CIRCLE NO. 187 ON READER SERVICE CARD

PC WEEK THE NEWS WEEKLY FOR IBM SYSTEM MICRO COMPUTERS

Complimentary Subscription Application

YOU MUST ANSWER ALL QUESTIONS TO QUALIFY.

1. ☐ **Please accept my request for a complimentary subscription to PC WEEK.**

2. ☐ **No, I do not want to receive PC WEEK.**

3. **Primary business activity of your firm at this location** (check one only):
   - ☐ 1. Manufacturing of computer equipment, computers, DP hardware, peripherals
   - ☐ 2. Manufacturing other
   - ☐ 3. Agriculture, mining, construction
   - ☐ 4. Transportation
   - ☐ 5. Communications
   - ☐ 6. Utilities
   - ☐ 7. Computer related retailer
   - ☐ 8. Non-computer related retailer or wholesaler
   - ☐ 9. Finance, banking, insurance, real estate
   - ☐ 10. Health, medical service
   - ☐ 11. Legal
   - ☐ 12. Education
   - ☐ 13. Data processing, computer service
   - ☐ 14. Other business services
   - ☐ 15. Government
   - ☐ 16. Consultant
   - ☐ 17. Other (please specify) ____________

4. **Your title:** (Check one appropriate box)
   - ☐ 1. Pres/Owner/Dir/Chmn/Partner
   - ☐ 2. Vice-Pres/Gen Mgr
   - ☐ 3. Comptroller/Treasurer
   - ☐ 4. Vice-Pres Finance
   - ☐ 5. Chief Accountant
   - ☐ 6. Vice-Pres Operations
   - ☐ 7. Dir/Mgr Purchasing
   - ☐ 8. Dir/Mgr Office Systems
   - ☐ 9. Vice-Pres Sales/Sales Mgr
   - ☐ 10. Dir/Mgr Marketing
   - ☐ 11. Dir/Mgr/ Chief of EDP/MIS
   - ☐ 12. Dir/Mgr Programming
   - ☐ 13. Dir/Mgr Systems & Procedures
   - ☐ 14. Dir/Mgr Communications
   - ☐ 15. Dir/Mgr EDP/MIS Operations
   - ☐ 16. Dir/Mgr Information Ctr
   - ☐ 17. Business Microcomputer Specialist
   - ☐ 18. Systems Analyst
   - ☐ 19. Data Base Administrator
   - ☐ 20. Vice-Pres/Dir R&D
   - ☐ 21. Vice-Pres/Dir Engineering
   - ☐ 22. Dir/Mgr Plant or Prod
   - ☐ 23. Chief Engineer
   - ☐ 24. Systems Engineer
   - ☐ 25. Design/Production/Research Engineer
   - ☐ 26. Consultant
   - ☐ 27. Educator
   - ☐ 28. Other (please be specific) ____________

5. **Do you have any *MAINFRAME* computers or *MINICOMPUTERS* on site at this location?**
   - ☐ Yes (If yes, please report accurately below for the two largest.)
   - ☐ No

Manufacturer	Model	Quantity

IBM or Compatible Personal (Micro) Computer Information for this location:
(Please report accurately for each model indicated)

Manufacturer's Name, Model	6. Currently Owned (Quantity)	7. Plan To Purchase Within: 0-12 Months (Quantity)	13-24 Months (Quantity)	25-36 Months (Quantity)
IBM PC				
IBM PC XT				
IBM XT/370				
IBM PC/3270				
IBM PCjr				
IBM Compatibles (Compaq, Eagle, etc.)				
Other (Not IBM or Compatibles)				

7a. **In which of the following ways are you yourself involved with the *Personal (Micro) Computers* at your location?**
   - ☐ A. Use them
   - ☐ B. Recommend them
   - ☐ C. Establish specifications
   - ☐ D. Approve purchase
   - ☐ E. Acquire them
   - ☐ F. Other Involvement (specify) ____________
   - ☐ G. No involvement

8. **The *Personal (Micro) Computers* purchased for this location would be:**
   - ☐ 1. For internal use
   - ☐ 2. For resale
   - ☐ 3. Other (please specify) ____________

Full company name and company address must appear in spaces below for application to be processed.

OR AFFIX YOUR BUSINESS CARD PW 1194

Name	
Title	
Company Name	
Division	Telephone Number
Company Street Address	
Company City	State / Zip

8a. **How are Personal (Micro) Computers normally obtained for this location?**
   - ☐ A. Direct from manufacturer
   - ☐ B. Manufacturers representative
   - ☐ C. Distributor
   - ☐ D. Retail computer store
   - ☐ E. System house
   - ☐ F. In-house (company) store
   - ☐ Z. Other (please specify) ____________

9. **Please indicate below the communications capability for which these *Personal (Micro) Computers* are used.**
   - ☐ A. Communicate with remote timesharing or database.
   - ☐ B. Communicate with internal mainframe or minicomputer.
   - ☐ C. Used in local area network.
   - ☐ D. Down load data from mainframe or remote service.
   - ☐ E. None of the above.

9a. **Please indicate below the applications for which these *Personal (Micro) Computers* are used.**
   - ☐ 1. Accounting
     - ☐ A. Accounts Payable
     - ☐ B. Accounts Receivable
     - ☐ C. Billing & Collection
     - ☐ D. General Ledger
     - ☐ E. Inventory
     - ☐ F. Order Entry & Invoicing
     - ☐ G. Time Billing
   - ☐ 2. Communications
   - ☐ 3. Data Base Management
   - ☐ 4. Data Input/Analysis
   - ☐ 5. Education
   - ☐ 6. Electronic Mail
   - ☐ 7. Entertainment (Games)
   - ☐ 8. Financial Planning
   - ☐ 9. Graphics Design
   - ☐ 10. Personal Time Management
   - ☐ 11. Portfolio Management
   - ☐ 12. Programming
   - ☐ 13. Project Management
   - ☐ 14. Process Control
   - ☐ 15. Scientific or Engineering Applications
   - ☐ 16. Statistical Analysis
   - ☐ 17. Tax Calculation or Planning
   - ☐ 18. Word Processing
   - ☐ 19. Other ____________

10. **Do you help acquire, recommend, specify or approve any of the products or services below?**
   - ☐ Yes (If yes, please check all that apply)
   - ☐ No.

Computers
- ☐ 1. Mainframe
- ☐ 2. Minicomputer
- ☐ 3. Personal (Micro)

Peripheral Equipment
- ☐ 7. Letter Quality Printer
- ☐ 8. Graphics Printer
- ☐ 9. High Speed Printer
- ☐ 10. Color Monitor
- ☐ 11. Monochrome Displays
- ☐ 12. Modems
- ☐ 13. Hard Disk
- ☐ 14. Tape Backup System
- ☐ 15. CPU Compatibility Card
- ☐ 16. Memory Board
- ☐ 17. Communications Port
- ☐ 18. Plotters/Charting Devices
- ☐ 19. Local Area Networks

Software Packages
- ☐ 21. Communications
- ☐ 22. Accounting
- ☐ 23. Order Entry/Inventory
- ☐ 24. Payroll
- ☐ 25. Time Billing
- ☐ 26. Financial Planners/Spreadsheet
- ☐ 27. Project Managers
- ☐ 28. Word Processors
- ☐ 29. Compilers
- ☐ 30. Database Managers
- ☐ 31. Program Developers/Generator Tools
- ☐ 32. Business Graphics

Outside Services
- ☐ 34. Maintenance
- ☐ 35. Education/Training
- ☐ 36. Software/Systems Design
- ☐ 37. Remote Computing
- ☐ 38. Database Services
- ☐ 39. Other

Expendables
- ☐ 41. Diskettes
- ☐ 42. Stock Paper
- ☐ 43. Forms & Other Consumables

11. **Are there any other individuals at this location that would qualify for a complimentary subscription to PC WEEK?**

Name____________________Title____________

Name____________________Title____________

Your Signature____________________

Your Title____________________**Date**____________

Please be sure to sign your name and list your actual title above.
See other side for mailing instructions.

SEAL THIS SIDE ONLY—DO NOT USE STAPLES

FOLD HERE

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY MAIL

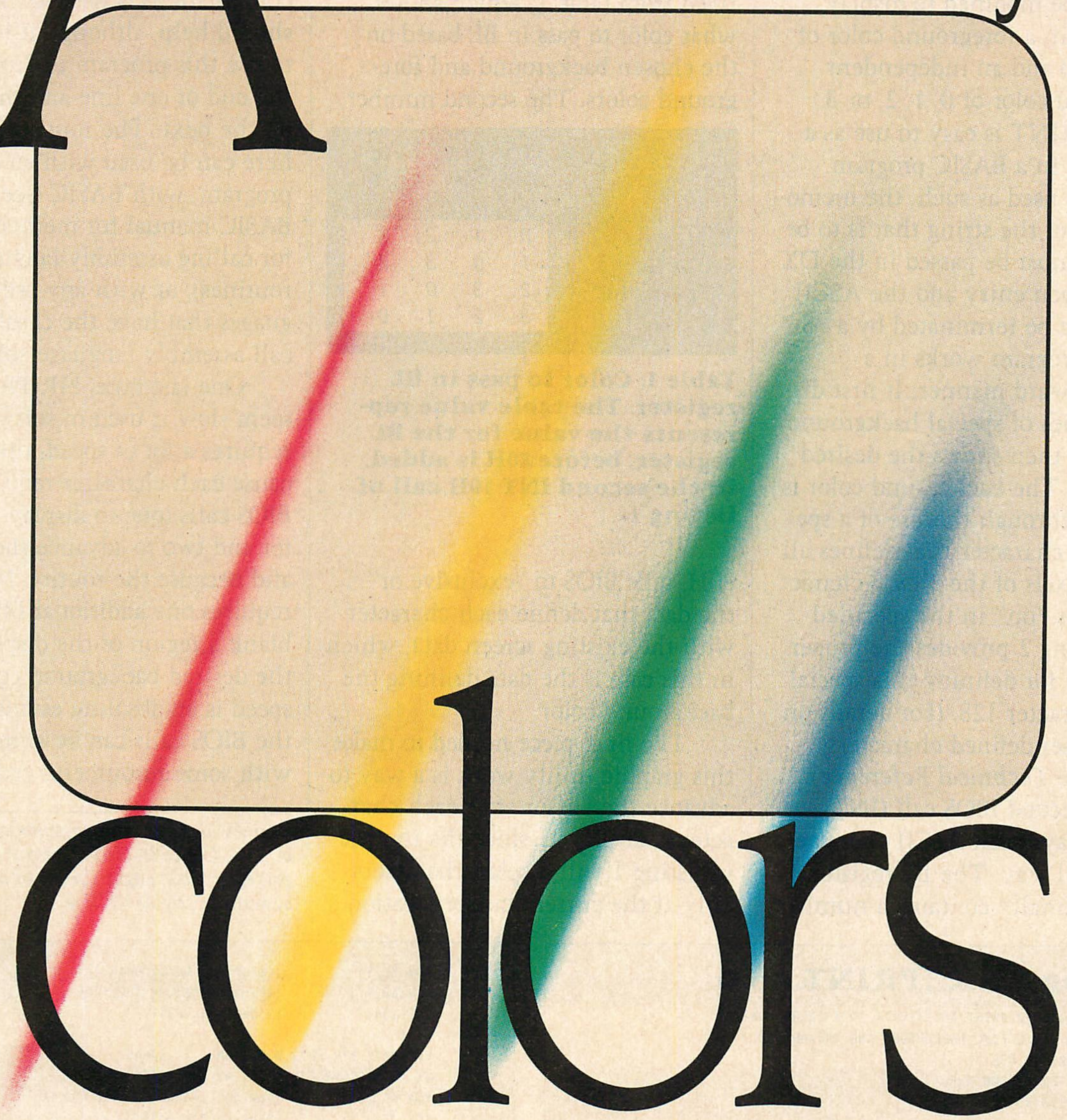
FIRST CLASS PERMIT NO. 3838 NEW YORK, NY

POSTAGE WILL BE PAID BY ADDRESSEE

PC WEEK
P.O. Box 5920
Cherry Hill, NJ 08034

DON AWALT

A screen of many colors



A routine for greater choice of character color in color graphics mode

MS-DOS and IBM BIOS provide routines to output a string of ASCII characters to a color display while in color graphics mode. MS-DOS provides function 9 (display a string until a termination character of "$" is reached), and BIOS interrupt 10 (hex) provides a routine to write an ASCII character at a given cursor location.

Although these routines provide some elementary (and general) aids to the graphics programmer, they offer little flexibility. One particular problem with them is that they always destroy any background that is not color 0 when a character is displayed.

A simple assembly language program can add flexibility to these routines by allowing the user to write ASCII characters to a graphics display in any of the four colors on a selected palette; a background color of any one of four colors can also be preserved.

Normally, ASCII characters are displayed on a color screen (mode 1, 320 × 200 graphics) in a foreground color of 3 and a background color of 0. The "NR_PRINT" routine in listing 1 can be modified to display characters in a foreground color of 0, 1, 2, or 3 and an independent background color of 0, 1, 2, or 3.

NR_PRINT is easy to use as a subroutine in a BASIC program. When it is used as such, the memory address of the string that is to be displayed must be passed in the DX variable upon entry and the ASCII string must be terminated by a "$."

The program works in a straightforward manner. It first displays a series of special background spaces and then shows the desired characters. The background color is generated through the use of a special ASCII character that defines all possible pixels of the 8 by 8 character block as "on" in the specified color. Listing 2 provides the proper subroutine for defining this special ASCII character 128. (For details on creating user-defined characters, refer to the *Technical Reference*).

The second BIOS call (INT 10H) displays each ASCII character in a special way. The BL register, which normally contains a number that represents a color to use in displaying the character, now contains the sum of two numbers. The first number needed to form the value of BL is a color number (0 to 3) derived from table 1, which shows what color to pass in BL based on the chosen background and foreground colors. The second number,

		Desired background color 0	1	2	3
Color of displayed character	0	0	1	2	3
	1	1	0	3	2
	2	2	3	0	1
	3	3	2	1	0

Table 1: Color to pass in BL register. The table value represents the value for the BL register, before 80H is added, in the second INT 10H call of listing 1.

80H, tells BIOS to "exclusive or" the data that define each character with the existing screen data, which in this case is the data defining the background color.

The final piece needed to make this graphic utility work is a way to advance the cursor after each character is displayed, since the routine in listing 1 will display characters only at the current cursor location. CURSOR_SET, shown in listing 3, uses BIOS interrupt 10H to advance the cursor by 1.

Sometimes the capability to display characters in unusual color combinations is needed. NR_PRINT should help, although it is difficult to use this program to display past the end of one line and then start on the next. The routines shown here can be used with an assembly program, with BASIC (see the BASIC manual for methods to use for calling assembly language subroutines), or with any other languages that have the capability to call assembly language subroutines.

One last note: NR_PRINT will seem slow if used in software that requires a lot of speed; this is because each character requires three BIOS calls (one to display a character and two to advance the cursor) and because the routine NR_PRINT requires one additional BIOS call to blank a region of the display with the desired background color. If speed is an absolute essential, all of the BIOS calls can be eliminated with some resourceful coding.

Don Awalt is systems engineering supervisor at EMC Controls. He is a former assistant professor who taught theory of operating systems at Loyola College.

Listing 1 NR_PRINT.ASM

```
;       ROUTINE TO PRINT ASCII DATA TO A COLOR MONITOR
;       USER CAN DEFINE FOREGROUND AND BACKGROUND
;       COLORS TO USE
;       WRITTEN BY DON AWALT
;       edit 12/83
;
DLR_SIGN EQU '$'            ;terminating character for
                            ;input string
SPCL_BLANK EQU 128
;
NR_PRINT PROC NEAR          ;input in dx, term by '$'
        MOV DI,DX           ;save in di for later use
        MOV SI,DX           ;save here also
        MOV CX,0            ;will tell nr of characters
                            ;to blank
GET_SIZE:
        MOV AL,[SI]         ;first (or next) char of string
        SUB AL,DLR_SIGN     ;compare with term character
        JZ BLANK_IT         ;if equal, found the $
        INC CX              ;count 1 more character
        INC SI              ;advance to next in string
        JMP GET_SIZE        ;go again
BLANK_IT:                   ;now to blank all positions
                            ;at once
        MOV BH,0            ;page nr for bios call
        MOV AL,SPCL_BLANK   ;defined in listing 2;
                            ;ascii 128
        MOV BL,1            ;color for background (1)
        MOV AH,9            ;write character function
        INT 10H             ;video bios call
P_NXT:
        CMP BYTE PTR [DI],DLR_SIGN      ;done?
        JE PRINT_COMPLETE   ;yes; we hit dollar sign
        MOV AL,[DI]         ;now get character to display
        INC DI              ;prepare for getting next char
        MOV BL,82H          ;color 2 (with background 1
                            ;gives color 3)
        MOV AH,9            ;write character function
        MOV BH,0            ;display page 0
        MOV CX,1            ;write 1 character
        INT 10H             ;call bios (video)
        CALL CURSOR_SET     ;listing 3; advance cursor
        JMP P_NXT           ;see if any more characters
PRINT_COMPLETE:
        RET                 ;done the work
NR_PRINT ENDP
```

Listing 2 Subroutine for Defining Special Characters

```
;edit: rda 12/83
;
;this list defines an 8x8 character with
;all pixels turned "on."  See BIOS listing
```

```
;in Tech Reference for other character definitions.
;
BLANK_DATA DB OFFH,OFFH,OFFH,OFFH,OFFH,OFFH,OFFH,OFFH
;
INIT PROC NEAR              ;this will set up an ASCII
                            ;character 128
        LEA DX,BLANK_DATA   ;all pixels on will be
                            ;ASCII 128
        MOV AL,1FH          ;address of our ascii chars
                            ;goes into vector 1fh
        MOV AH,25H          ;dos fcn call: set interrupt
                            ;vector
        INT 21H             ;dos fcn call
        RET                 ;note that this must be done
                            ;before nr_print!
INIT ENDP
```

Listing 3 CURSOR_SET.ASM: Routine to Advance the Cursor

```
;edit rda 12/83
;
CURSOR_SET PROC NEAR   ;advance cursor by 1
        MOV AH,3       ;bios read cursor position fcn
        MOV BH,0       ;display page 0
        INT 10H        ;dx has row,column of cursor
        INC DL         ;this increments column
        MOV AH,2       ;write cursor fcn call
        MOV BH,0       ;display page 0
        INT 10H        ;write cursor
        RET
CURSOR_SET ENDP
```

TECH NOTEBOOK 25

MORE without Redirection

SHOW.COM eliminates the input redirection symbol from MORE.COM.

MURRAY LESSER

The sample filter MORE.COM (distributed with the diskettes for PC DOS 2.x) could be a very handy substitute for the DOS command TYPE, because it allows a user with a fast hand on the Ctrl-NumLock key to keep the displayed ASCII file from scrolling off the screen. MORE suffers from one grievous fault, however: while it is useless without a target file to display, it requires the user to insert the redirection symbol before it will perform. If the user forgets the redirection symbol, MORE just sits there displaying everything that is keyed in—twice—until it is killed with a Ctrl-Z or Ctrl-Break.

Fortunately, the necessity for redirection can be removed with a little patching. But before a proprietary program can be patched, it has to be disassembled so the user can figure out how it works, what changes need to be made, and where they should go. All of this can be done with DEBUG. Such disassembly of short proprietary programs is also a good way to learn some of the tricks of the trade in using the DOS facilities that are not explained clearly in the manual.

Disassembling MORE.COM showed that the code for "unredirecting" the standard input and storing the replicated "handle" for the target file in register BP is located in offsets 0128H through 0133H. MORE reads the target file (in four-kilobyte chunks) into memory starting at offset 021AH. The space in the original file between 021AH and the end-of-file at 027FH is filled with <NUL> [00] bytes.

Thus, the patch to delete the requirement for redirection begins with a JMP command at 0128H, jumps to new code at 0220H, and ends with a jump back to 0134H.

The procedure for patching MORE.COM (after copying it to SHOW.COM to avoid confusion) is shown in figure 1. The keyed entries are in bold type. All numbers displayed or used by DEBUG are in hex. The segment values in the figure are for my system and probably will be different for other systems. C drive is a virtual disk that is part of resident DOS.

SHOW can be used as MORE was, except that the input redirection symbol can be omitted from the command line. If the user gives SHOW an input redirection symbol, it will respond as if a valid <filespec> had been left off of the command line with the DOS prompt. If the redirection symbol and an invalid <filespec> are given, the command processor will return a "File not found" message before it shows its prompt.

Murray Lesser is the author of Using Microsoft Compiled BASIC, *published by McGraw-Hill in September. He retired after 26 years at IBM, where his last job was program manager, technical documentation.*

Figure 1: Patching SHOW.COM with DEBUG

```
C>DEBUG SHOW.COM
-R
AX=0000  BX=0000  CX=0180  DX=0000  SP=FFFE  BP=0000  SI=0000  DI=0000
DS=378B  ES=378B  SS=378B  CS=378B  IP=0100   NV UP DI PL NZ NA PO NC
378B:0100 B430          MOV     AH,30
-A128                           ["Assemble" starting at offset 0128]
378B:0128 JMP 220               [Jump to patch in offset 0220]
378B:012B                       [Exit "Assemble" mode with <Enter>]
-A220                           ["Assemble" patch starting at 0220]
378B:0220 XOR AH,AH             [Put count of number of bytes in
378B:0222 MOV SI,80             [ "rest of command line"
378B:0225 LODSB                 [ into AX register]
378B:0226 OR AL,AL              [Anything there?]
378B:0228 JNZ 22C               [If so, continue]
378B:022A INT 20                [Else terminate program]
378B:022C MOV DI,81             [Set DI register to byte following
378B:022F ADD DI,AX             [ "rest of command line",
378B:0231 XOR AL,AL             [ turn off Carry Flag, and convert
378B:0233 STOSB                 [ <filespec> to ASCIIZ string]
378B:0234 MOV AH,3D             [Open <filespec> for reading,
378B:0236 MOV DX,82             [ returning "handle"
378B:0239 INT 21                [ in AX register]
378B:023B JC 22A                [Terminate if Carry Flag shows error]
378B:023D MOV BP,AX             [Move handle to location MORE expects
378B:023F JMP 134               [ and return to original code]
378B:0242
-W                              [Write patched SHOW.COM to file]
Writing 0180 bytes
-Q                              [Exit DEBUG]

C>

Notes: Keyed input has been
emphasized and comments in
square brackets added.
```

Beyond Board-Driven Networks

Multi-User PC Networks. Their ability to share expensive peripherals, programs, and files has always made sense. At least, in theory. The catch is that, previously, every workstation required its own PC . . . PLUS a kilobuck "Network Interface Board." In other words, your cost-per-user was over $3,000. And that doesn't make sense. Not any more.

The Multi-Tasking, Multi-User Network That's Software-Driven. MultiLink™ represents a major breakthrough in networking systems. Network interface boards are not required. In addition, only one IBM PC/XT is needed in your network.

Four Workstations for the Price of One. MultiLink™ is a unique software package which transforms ONE IBM PC/XT (or PC-Compatible) into the CPU of a multi-tasking, multi-user network. It utilizes inexpensive terminals as workstations. Up to eight can be connected to a single PC with appropriate serial ports. Currently, four of these terminals cost less than one PC.

MultiLink™ runs under all releases of PC-DOS (except 1.0) and certain implementations of MS-DOS. Leading programs such as WordStar, Lotus 1-2-3, PeachText 5000, and dBASE II are supported. In addition, most text-oriented software is fully-compatible.

A Resource-Sharing System. Multiple users are able to access all of your computer's resources, locally, or from remote locations using a terminal with a modem. Disk, printer, program-sharing, and a comprehensive print spooler are fully-implemented.

Make the MultiLink™ Connection Today. Call The Software Link TODAY, and get the whole story. MultiLink™ is immediately available for $295 and comes with a money-back guarantee. VISA, MC, and AMEX accepted.

MultiLink™

THE SOFTWARE LINK, INC.

8601 Dunwoody Place, Suite 336, Atlanta, GA 30338 Telex: 4996147 SWLINK

CALL: 404/998-0700

Dealer Inquiries Invited

MultiLink™ is a trademark of The Software Link, Inc. Terminals courtesy of QUME Corporation.

IBM, PC, XT, & PC-DOS are trademarks of IBM Corp. MS-DOS, WordStar, dBASE II, Lotus 1-2-3, and PeachText 5000 are trademarks of Microsoft Corp., MicroPro, Ashton-Tate, Lotus Development Corp., & Peachtree Software, Inc., respectively.

CIRCLE NO. 198 ON READER SERVICE CARD

IF "C" IS YOUR LANGUAGE,

LATTICE WINDOW™

From Now On, All Your Applications Could Have Windows!

NEW RELEASE!

Windows are no gimmick. Ask any user of a windowed product.

Windows can bring unrehearsed combinations of data side-by-side for viewing. Windows can overlay new information atop old, yet can restore the old with a keystroke. Windows let the limited space of a screen show many things, not just one thing.

Windows are one of the new software features people have come to expect. And why not? They make using software more convenient, more versatile. But how do you add this technique to your applications without disappearing for several months of R&D?

With Lattice Window, that's how. It's a product we've been waiting months for. One of those rare programming tools which will change the way you think of program design. Here's how:

Lattice Window takes over all screen management. It is an extensive set of object code functions which you merely call from your C program.

Think of a window as a screen, a screen of flexible size. You can tell Lattice Window to open and close many such virtual screens. Up to 255 at a time, each from 1x1 to 255x255 bytes. Then tell "Window" to display any portion of these virtual screens on the physical screen — as many as fit. Tell "Window" to place them wherever you want, overlapping and overlaying at will.

Windows no longer needed may be closed. Any background area overlaid earlier (which could be corners of several windows) will pop back to the screen. Think for a moment what a programming job that technique represents.

To your program, the entered window is the *entire* screen — all row and column references are relative to the window no matter its absolute position in the screen. All scrolling occurs only within the active window. The cursor will not leave its boundaries until your program says to. Any screen window may be entered for display, data entry, whatever. *Any* virtual screen — displayed or not — may be read from or written to by your program.

- You can move a window on the screen.
- You can grow it to display more of the underlying virtual screen. Or shrink it to show less.
- You can control screen attributes (color, blinking, inverse video, etc.) of any window.
- You can direct output to either the monochrome or color board.

It's a formidable product which will add incomparable flair to your next application!

Product Code: S0800
Suggested Price: **$295.00**
Our Price: **$235.00**

HALO

A Spectacular Graphics Extension to Lattice C

Halo™ will astound you. It provides a complete library of graphic functions which can be linked with your Lattice programs to create full-color charts, graphs, simulations, even animation.

Over 100 commands are at your disposal, including plot, line, arc, box, circle, plus single commands to produce bar graphs and pie charts. Pattern-fill and dithering commands give your graphics impressive texture and color mixes, rubber-banding draws shapes for interactive users, area moves produce animation, fill and flood commands paint areas. The newest version allows you to define your own world coordinate scheme, divide the screen into "viewports" (windows), and scale graphic figures automatically.

It's a long list of capabilities which make for an extraordinarily powerful product. In fact, Halo is so good that manufacturers of graphics boards and systems are adopting it as a standard graphics language. So it can bridge your application to other systems. CAD-CAM developers, especially, have embraced its device-independent approach for maximal portability.

Halo is a dazzling demonstration of why C has become the language of choice among programming professionals: its function library architecture means you can tremendously enhance your firepower by acquiring libraries of software like Halo with dramatic economy of time and money.

Requires IBM monochrome or color graphics card or equivalents.

Product Code: S0300
Suggested Price: **$200.00**
Our Price: **$125.00**

PMATE

The Programmer's Word Processor

Pmate was designed for programmers. We'll wager that you cannot find a programmer who has discovered Pmate and moved on to something else.

Pmate is a full screen editor with ten auxiliary buffers for squirreling away pieces of text until needed. It uses single key commands to move the cursor, or text, or insert or delete, or rescue several thousand characters of deleted text.

It has a format mode for tab setting or wraparound and shaping when it's time to write documentation. Pmate lets you assign chains of commands or strings of text to single keys: one keystroke could set up the entire shell of a new C function, for example.

Pmate has variables, if-then statements, loops. It calculates, and converts decimal to hex to binary and back. You can write compact programs (called "macros") to delete comments, for example, or check syntax, or process long sequences of commands. Macros can alphabetize lists, do row and column math, perform a series of operations on multiple files, even summon other macros.

Put another way, Pmate is a text editor with its own built-in interpretive language. A language you can use to completely customize this text editor to your fancy. Possibly the most artful, ingenious program you have ever seen.

Product Code: S0600
Suggested Retail: **$225.00**
Our Price: **$175.00**

PLINK86

Overlay Linkage to Expand

Software is becoming ever more sophisticated, which means more complex programs requiring large chunks of memory. But if you use extra memory, if you count on users to have expanded RAM, you will forego sales to those who do not.

Plink86 is the answer. It takes on the job of shoe-horning large programs into small memory. First, Plink86 acts as an alternative to DOS' Link. For a language like C which encourages design of separately compiled object modules in the Microsoft relocatable format, Plink86 pulls modules together into single compiled programs. But Plink86's overlay power is what has gained it a reputation as a miracle worker. It binds into the compiled program its overlay manager which knows how to swap modules of your large linked program between disk and memory, so that each can temporarily occupy the same memory space.

Unlike other linkers, the overlay manager acts on its own, needing no calls from the source program. Instead, Plink86's straightforward overlay description language allows you to describe your overlay structure in one place in your program — a structure

PC BRAND HAS THE TOOLS

Your Productivity Could Take Off With Our Craftsman™ Line of Precision Tools

C-FOOD SMORGASBORD

A Menu of Difficult to Prepare C Functions

C is a language of function libraries: sturdy routines which your application can effortlessly reference to get the job done. But someone has to write them. It would take months to concoct and perfect the complex and useful servings which have been prepared for you and thoroughly tested in C Food Smorgasbord™.

Decimal Arithmetic: Functions to perform operations on binary-coded decimal (BCD) numbers of up to sixteen significant digits. Includes trigonometric and logarithmic functions, powers, conversions to strings, and much more.

Level 0 I/O Functions: Enable you to perform direct I/O operations for screen, keyboard, printer, and asynchronous port with no dependence on higher I/O functions to minimize memory usage and maximize speed.

IBM™ PC BIOS Interface Access: Allows you to get at the basic I/O services in the ROM BIOS not available through the normal operating system to get and set the video mode, cursor position, color and screen attributes, keyboard shift, scrolling, printer channel and port status.

TIP, the terminal independence package: Lets you easily move programs to computers with different types of terminals.

There is a basket of delicacies beyond this brief list; a cornucopia to sample, any one of which will save valuable time and pay back far more than C-Food Smorgasbord's overall price.

Product Code: S0200
Suggested Price: **$150.00**
Our Price: **$115.00**

All products operate on the IBM PC or PC/XT under PC-DOS and require no more than 64K of RAM unless indicated.

Prices, terms and specifications subject to change without notice.

Prices do not include shipping and handling.

You may freely use these products to create software applications for re-sale without additional licenses or royalty payments, provided your applications are not wholly or substantially replications of the products themselves.

Unix TM Bell Laboratories / Lattice C, C-Food Smorgasbord, and Lattice Window TM Lattice Inc. / MS TM Microsoft Inc. / Halo TM Media Cybernetics / IBM TM International Business Machines / Float87 TM Microfloat / Panel TM Roundhill Computer Systems Ltd. / Plink86 and Pmate TM Phoenix Software / CP/M-86 TM Digital Research

Your Art of the Possible

permitting up to 4,095 overlays stacked 32 deep. And you don't have to re-compile to re-arrange the structure.

Plink86 is a two pass linkage editor. On the first reading it determines all modules which need loading, to insure greater flexibility in assigning memory segment addresses before the disk file is created on the second pass. It can even sub-divide its linked output into multiple files for programs which must span more than one disk.

But most of all it sets you free to write the comprehensive code today's users have come to expect without sacrifices to memory constraints.

Product Code: S0500
Suggested Price: **$395.00**
Our Price: **$315.00**

PANEL

Are You Still Coding Hand-Made Data Entry Screens?

Designing custom screens is intricate, time-consuming, and costly. It can make you the loser in competitive bidding. It can add weeks, even months to bringing your product to market.

But not when you use Panel™. Put Panel to work and you hire a superprogrammer for the minimum wage. Telling Panel your screen design is like using a word processor. You can add, move, copy, delete, and reshape fields with single key commands. The result? Finished C program code with field descriptors, color and highlighting, input editing by type of field, help keys, and warning messages.

Panel can also create a terminal selection program customized for your application, and gives you a utility to quickly test your finished screens. It even includes a multi-key data file maintenance program which interacts with the screen you design. All the tools you need to generate code for the trickier aspects of your application, or even to create stand alone data entry and retrieval products without *any* programming.

Panel routines are powerful. Unlike so many programs which make error correction difficult once a field is departed, Panel gives the user full field-to-field movement for editing, and overtype or insert/delete within fields.

A truly superior productivity tool every developer should have.

Product Code: S0400
Suggested Price: **$295.00**
Our Price: **$235.00**

FLOAT-87

Software Support for the 8087 Math Chip

Float-87™ is a library of floating-point math routines which are merged into a Lattice C interface library. The routines handle all arithmetic functions and the most frequently used transcendental or trigonometric functions, and pass all numbers to the 8087 to take advantage of its 80-bit data format.

Putting Float-87 to work with an 8087 can increase the speed of floating-point calculations by 40 times or more — and with greatly expanded accuracy. If you or your customers have an Intel 8087 math processor chip on board, this software will switch on its afterburners.

Product Code: S0700
Suggested Retail: **$125.00**
Our Price: **$100.00**

You won't find an offer like this elsewhere: Exceptional prices and software fully supported by a knowledgeable staff specializing in 'C' and companion tools.

1-800-PC BRAND

That's 1-(800) 722-7263. In NY State call (212) 410-4001.
Charge your purchase to MasterCard or Visa.
Or mail your order and check to
PC Brand, P.O. Box 474, New York, N.Y. 10028

CIRCLE NO. 170 ON READER SERVICE CARD


dBASE III™
dBASE III™
SYSTEM DISK
Serial No: 100001
Format: IBM PC, XT and 100% compatibles
PC-DOS 2.0
Version: 1.0
ASHTON·TATE
10150 West Jefferson Blvd.
Culver City, CA 90230 (213) 204-5570
TELEX 664228 ASHTATE CULV
©Copyright 1984 Ashton-Tate
as an unpublished licensed
proprietary work.
All rights reserved.

More power to you.

Remember the magic you expected when you first purchased a PC?

It's here.

dBASE III™ is the most powerful database management system ever created for 16-bit microcomputers. It pulls every ounce of energy out of your PC and puts it to work.

On top of that, it's fast and it's easy.

You've never seen anything like it.

dBASE III can handle over a billion records per file, limited only by your computer system. You can have up to ten files open, for sophisticated applications programs.

When you have two related files, information in one can be accessed based upon data in the other.

dBASE III now handles procedures, parameter passing and automatic variables. You can include up to 32 procedures in a single file. With lightning speed. Because once a file is opened, it stays open. And procedures are accessed directly.

Easier than ever.

dBASE III uses powerful yet simple commands that are the next best thing to speaking English.

If you're unsure of a command, HELP will tell you what to ask for.

If you don't know what command comes next, a command assistant does. All you have to know is what you want it to do.

Our new tutorial/manual will have you entering and viewing data in minutes rather than reading for hours.

And to make matters easier, you get a full screen report setup for simple information access.

Faster than no time at all.

dBASE III isn't just fast. It's ultra-fast. Operating. And sorting. Even faster, is no sorting. Because dBASE III keeps your records in order, so you really don't have to sort anything. Unless you want to. Then watch out!

What about dBASE II®?

It's still the world's best database management system for 8-bit computers. And it's still the industry standard for accounting, educational, scientific, financial, business and personal applications.

Tap into our power.

For the name of your nearest authorized dBASE III dealer, contact Ashton-Tate, 10150 West Jefferson Boulevard, Culver City, CA 90230. (800) 437-4329, ext. 333. In Colorado, (303) 799-4900.

CIRCLE NO. 300 ON READER SERVICE CARD

ASHTON·TATE™

INSIDE 1-2-3 WORKSHEET FILES

William F. Sharpe

It's possible to create a program that takes advantage of 1-2-3's good points while lessening its shortcomings

Since it appeared on the market Lotus 1-2-3 has maintained its position on the best-seller lists of programs sold for the IBM Personal Computer—and apparently for good reason. Seasoned PC operators seem able to do almost anything with it. Nothing is perfect, however.

For some tasks 1-2-3 requires substantial effort and has an excessive "run time." It is difficult (but not impossible) to do multiple regression, quadratic programming, and matrix inversion with 1-2-3, although these tasks are more easily performed with other programs.

Even for such procedures, however, 1-2-3 is ideal for preparing input, transforming variables in various ways, and providing graphs of output. Its good points can be used to advantage and its drawbacks alleviated if 1-2-3 is used for preparation of input for a "foreign" program and for analysis of its output.

A typical procedure involves five steps:

1. Prepare input using 1-2-3, saving the input as a 1-2-3 worksheet file.
2. Use the Lotus Translate facility

William F. Sharpe is Timken Professor of Finance at the Stanford University Graduate School of Business. He currently is preparing a series of programs under the trademarked name "Worksheet Tools."

to convert the 1-2-3 worksheet file to a DIF file.

3. Run the foreign program, which takes its input from the DIF input file and sends its output to another DIF file.
4. Use the Lotus Translate facility to convert the DIF output file to a 1-2-3 worksheet file.
5. Load the output worksheet file into 1-2-3 for further analysis, plotting, etc.

This is not simple. Moreover, since most of the information is numeric, a great deal of translation is involved. Lotus 1-2-3 stores numeric data in binary format. In the DIF format, numbers are stored as ASCII characters. Thus step two translates binary numbers to ASCII character strings. The foreign program must retranslate the character strings to binary numbers, do its work, then translate its numeric output to character strings to be written in the DIF format. Step four involves further translation—from character strings to binary numbers. For large problems, such translation can require minutes of processing.

Happily, there is a better way. An analytic program can read input directly from a 1-2-3 worksheet file and prepare another 1-2-3 worksheet file containing output. This obviates steps two and four, speeds up the process, and allows more information to be transferred between 1-2-3 and the "foreign" program.

To write such programs, of course, it is necessary to understand the format of 1-2-3 worksheet files. Lotus does not currently provide this information, but it can be deduced by examining enough examples of files created with 1-2-3.

The following information is believed to be accurate for Version 1A of Lotus 1-2-3 for the IBM Personal Computer. Other implementations may differ substantially. A Microsoft (IBM) BASIC program for reading and printing information from a file will be used for illustration. BASIC is a particularly inappropriate language for this task, but it is familiar and widely available. A more efficient (but more complex) BASIC program could be prepared. Better yet, a more appropriate language (for example, Pascal or C) could be selected.

THE RECORD STRUCTURE

Worksheet files are composed of a series of records of variable lengths. Each record begins with a header, consisting of a pair of two-byte integers that indicates the record type and length. The record length indicates the number of bytes in the remainder of the record. Figure 1 shows the general structure of a worksheet record.

> *For some tasks 1-2-3 requires substantial effort and has an excessive "run time."*

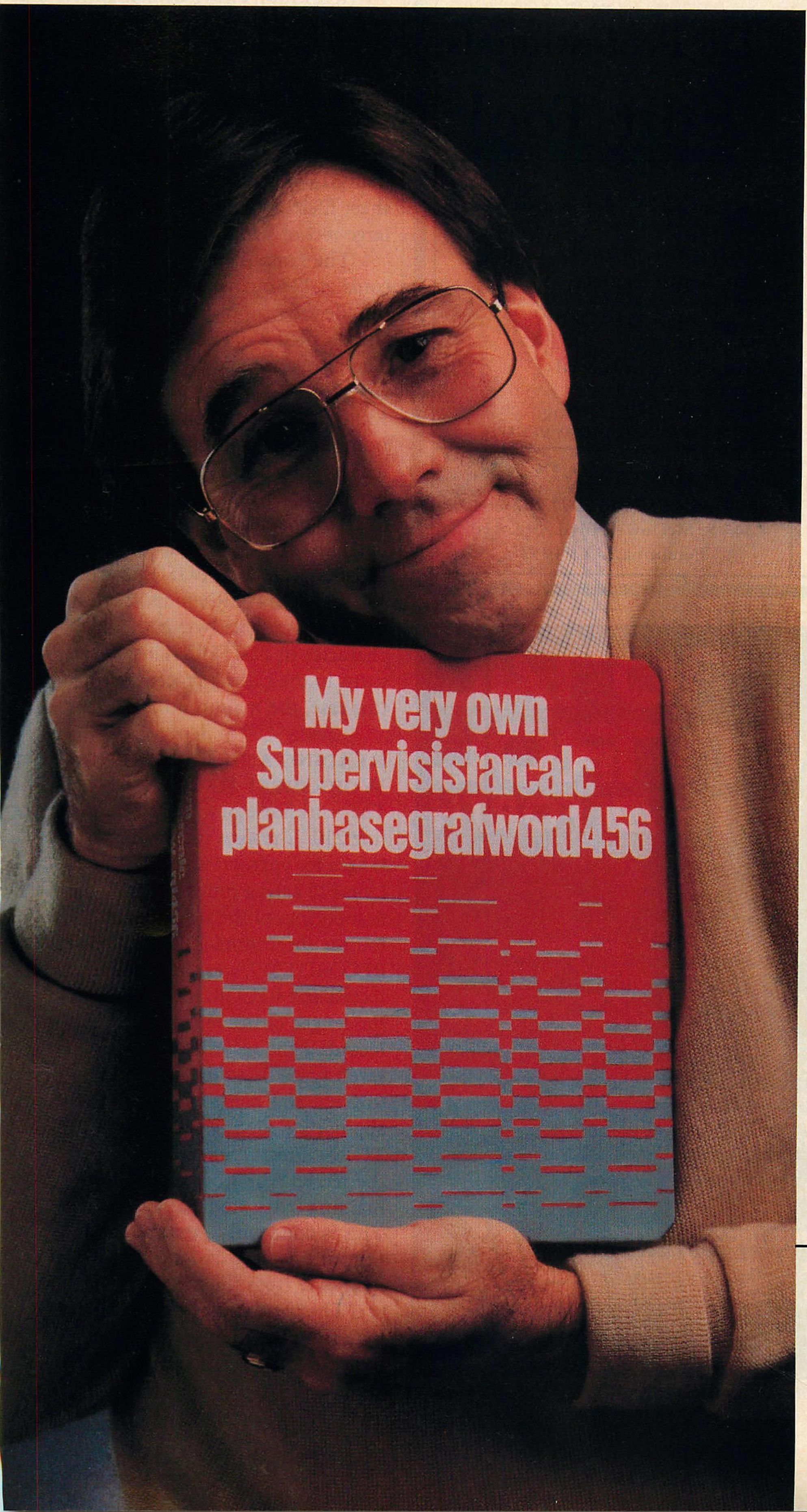
There are many types of records. For reading and writing files of data, the following suffice:

- Record type 0 : Header
- Record type 6 : Range
- Record type 13: Integer value
- Record type 14: Double-precision value
- Record type 15: Character string (Label)
- Record type 16: Formula and current value
- Record type 1 : End-of-worksheet

Examples of each of these types of records are provided in figure 2. Listing 1 gives the BASIC program to read and print the contents of a worksheet file.

To process a worksheet involves: opening the file and checking for a valid header; reading and processing records; and stopping when the end-of-worksheet record has been read. Lines 40 through 100 of the program provide a "main routine" to do this. Two routines are called—one (line 120) to open the

Do it yourself!



When developing effective applications for the PC is a matter you want to take into your own hands, first get your hands on the APL★PLUS®/PC System from STSC. This unique application development tool gives you the power to deliver totally integrated, highly efficient solutions to problems that conventional software simply can't handle.

The APL★PLUS/PC System is the most advanced application development system available today. And it includes enhancements for the professional, hobbyist, or novice programmer.

—Advanced *screen management facility* with windowing and scrolling features.

—Complete *on-line HELP facility* for novice users.

—Powerful *communication capability* for moving data between computers.

—Comprehensive *color graphics capability, and much more.*

Because of the efficiency of the APL★PLUS/PC System, you can develop programs 4 to 10 times faster than with conventional languages like BASIC or FORTRAN. And the system can be integrated with other applications you've already bought, or used to link those that aren't speaking to each other. It doesn't demand that you abandon those skills you've learned or the software you've come to love. But it does enhance your programming power, so you can do what you do even better!

Get the advanced capabilities of the APL★PLUS/PC System for only $595.00, a valuable investment when you can have things done your own special way. To order the APL★PLUS/PC System see your local dealer or contact STSC, Inc., (800) 592-0050, 2115 East Jefferson Street, Rockville, Maryland 20852.

CIRCLE NO. 208 ON READER SERVICE CARD

STSC
A Contel Company

The APL★PLUS/PC System runs on the IBM PC with 192 KB of RAM, as well as on a number of compatible machines.

file, and another (line 210) to read and process a record.

Worksheet files may contain anything—including a Control-Z (ASCII 26) character. Since this signals the end of a standard text file, "random" file access must be used, even for serial processing. In BASIC, random files may have no more than 32,767 records. Since worksheet files can be long, a relatively large record size is necessary.

The sample program uses records of 128 characters. Portions of the file are read into a character string called BUFFER$, as needed. The procedure to do this extends from lines 120 through 190. Lines 130-160 receive the name of the file from the user, append .WKS, open it as a random file with a record length of 128, and assign all input to a 128-character string named BUFFER$. Line 170 reads the first 128 characters into BUFFER$. Line 180 sets POINTER% to indicate that the next available byte is the first character in BUFFER$. Line 190 returns to the calling procedure (in this case, the main routine).

READING AND PROCESSING WORKSHEET RECORDS

The workhorse procedure begins at line 210. It reads a record and arranges for appropriate processing. Line 220 calls a routine to read the first portion of the next record. This routine (beginning at line 370) sets RECORDTYPE% to the type of record and RECORDLENGTH% to the length of the remainder of the record. If the record is one of the seven types listed in lines 240 through 300, control is passed to the appropriate routine (which will RETURN directly when through). If the record is another type, the remaining bytes are read but ignored. This is done with a routine (beginning at line 480) designed to get the next byte from the file.

The basic strategy for getting a byte from the file uses a pointer, indicating the position in BUFFER$

in which the desired byte is stored. This is incremented by one after each byte is used. If the resulting value exceeds 128, the next 128 characters are read into BUFFER$, and the pointer is reset to 1.

Lines 480 through 550 do the work. The next byte is copied into BYTE$ as a character and the pointer is incremented so that it indicates the next character in BUFFER$. If necessary, a new set of bytes is read into BUFFER$ and POINTER% is reset. Line 490 saves the previous byte for possible further processing.

As indicated earlier, each record starts with a header—two bytes indicating the type of record, followed by two bytes indicating the length of the remainder of the record. These values are to be stored in two variables: RECORDTYPE% and RECORDLENGTH%. The requisite procedure begins at line 370. In each case, two bytes are obtained (using the procedure at line 480). They are combined to form a string (called PREVIOUSBYTE$+BYTE$) that then is converted to an integer value by the CVI function.

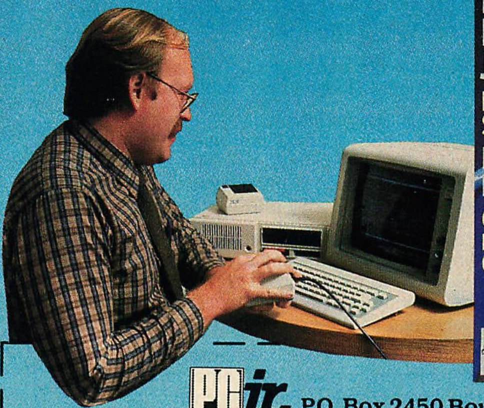
Processing the header (type 0) record is not strictly necessary. However, checking its validity is wise. It also should be the first record on the file. A valid header record has a length of two bytes. Version 1A of Lotus 1-2-3 writes an ASCII 4 in each one. All three conditions are checked in the procedure that runs from line 1010 to 1100.

DATA RECORDS

Data are stored in four types of records: integer value records (type 13), double-precision value records (type 14), character string records (type 15), and formula records (type 16). One such record exists for each occupied cell in the worksheet. No information is stored for empty cells.

Each data record includes information about location of the cell, including both row and column numbers. These row and column numbers begin with zero; thus, row

MATCH WITS WITH YOUR ACCOUNTANT, TEACHER ...OR ALIEN WARRIORS



Balancing your monthly statements or the fate of the universe can be equally harrowing experiences.

MASTER THESE CHALLENGES WITH PC*jr.*, The Independent Guide To The IBM PCjr in Education, Home and Business. PC*jr.* will help you fight off marauding invaders from other worlds AND help you maximize the performance of your PCjr with new business, education and finance applications!

Written in plain english, PC*jr.* is the guide you need to help you add-on new peripherals and modems ... to keep abreast of the newest releases in software and hardware ... *to make your PCjr a formidable business, educational and recreational computer!*

Subscribe to PC*jr.* and save! Save your finances! Save your grade point average! Save the universe!

SAVE UP TO 44% WITH PC*jr.*!

PC*jr.* P.O. Box 2450 Boulder, Colorado 80322

YES, please enter my subscription to PC*jr.* for:

☐ 8 issues for $9.97. I SAVE 40%!	☐ One year for $14.97. I SAVE 40%!	☐ Two years for $27.97. I SAVE 44%!

Savings based on full one-year subscription price of $24.97.

8Z011

Mr./Mrs./Ms. ______________________
Please print name in full

Address ______________________

City ______________ State ________ Zip ________

Check one: ☐ Payment enclosed ☐ Bill me later

Charge my: ☐ American Express ☐ Visa ☐ MasterCard

Card No. ______________________ Exp. Date ________

Offer Valid In U.S. And Possessions Only. Please allow 30 to 60 days for delivery of first issue.

number 3 is displayed as row 4, column number 0 is displayed as column A, etc. Cells are stored in order, beginning with the upper left portion of the worksheet, proceeding from left to right within rows and from the top row to the bottom.

When a portion of a worksheet is saved (for example, with a /File Extract Value instruction), cell locations are converted to those that would have been applicable if the portion had been in the upper left corner of the worksheet. For ease of interpretation, a cell's location can

Processing the header (type 0) record is not strictly necessary. However, checking its validity is always wise.

be printed in "external" terms—that is, as it would be displayed. Lines 680-760 contain a procedure designed to show the location of a cell in ROW% and COLUMN%.

In each of the four types of data records—13, 14, 15, and 16—the first byte indicates the format to be used when displaying the contents of the cell, and the next four bytes indicate the column and row numbers in which the cell is located. The procedure between lines 570 and 660 processes the first five bytes of a data record. After this has been run, the procedure beginning at line 680 can be used to print the row and column in "display" terms.

The simplest data record is type 13, which contains an integer value stored in two bytes. The location and contents of such records are obtained and printed with the procedure between lines 1340 and 1410.

Numeric data not stored as two-byte integers are stored as eight-byte (double-precision) floating-point numbers, such as are present in records of type 14.

Unfortunately, double-precision numbers are stored in one format by

1-2-3 and in another by Microsoft BASIC. All double-precision numbers in worksheet files are stored in the format used by the Intel 8087 coprocessor. The eight bytes representing such values are stored "backwards" on the file. To be used in programs written in Microsoft BASIC, such values must be converted. Some compilers for other languages (for example, Pascal and C) do not suffer from this drawback.

It is possible to call up an assembly-language routine designed to perform such a conversion from a BASIC program. Details can be found in Richard Startz's book, *8087 Applications and Programming for the IBM PC and Other PCs* (Brady, 1983). See listing 1, in which a procedure in Microsoft BASIC (beginning at line 780) is used.

The procedure to convert the double-precision value is rather inelegant. Lines 790 through 810 test to see if the value is not available (displayed by 1-2-3 as NA). In worksheet files such values are indicated by an ASCII 255 byte followed by an ASCII 240 byte. The remaining bytes have values of zero, but the first two suffice for purposes of identification. Line 820 and 830 test for a zero value. This is indicated by zero values for all eight bytes, but inspection of the first two suffices. The remainder of the procedure performs the conversion.

Double-precision value records are processed beginning at line 1430. Lines 1460 through 1490 get the eight bytes containing the value, putting the numeric values of the bytes in vector BYT%. The bytes are put in "backwards"—the value of the first in BYT%(8), the second in BYT%(7), etc.

When vectors BYT%(1) through BYT%(8) are ready, the conversion procedure is called. The result is returned as DOUBLE# (a double-precision number) unless ISNA% equals 1. In the latter case, the cell contains an NA. Line 1510 prints either the value or NA.

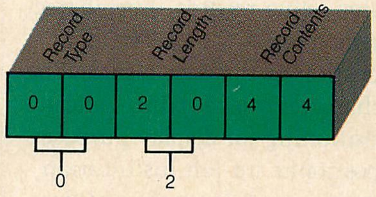


Figure 1: General Structure of a Worksheet Record

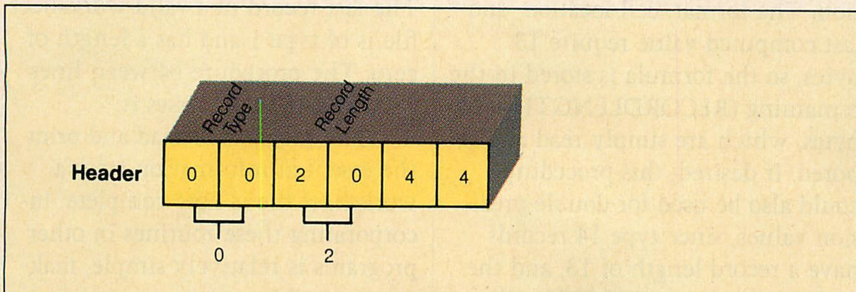
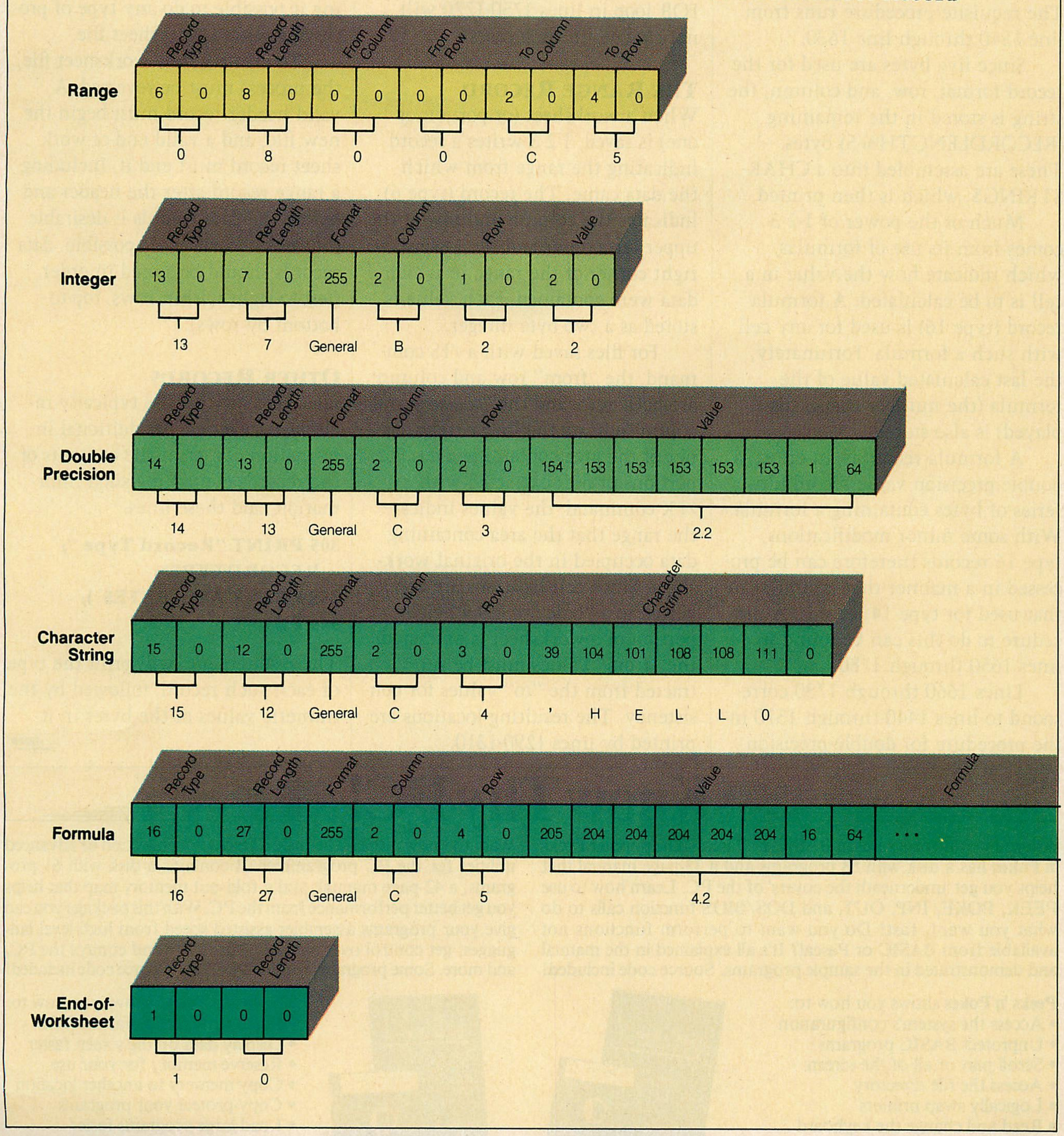



Figure 2: Examples of Worksheet Records

Character string records (type 15) are much simpler than double-precision value records. The character string follows the cell format and location in the data record. The first character of the string indicates the positioning for display purposes (an apostrophe for left justification, a quotation mark for right justification, a caret for centering, etc.). The last character is an ASCII 0 (null). The requisite procedure runs from line 1540 through line 1630.

Since five bytes are used for the record format, row, and column, the string is stored in the remaining (RECORDLENGTH%-5) bytes. These are assembled into a CHARSTRING$, which is then printed.

Much of the power of 1-2-3 comes from its use of formulas, which indicate how the value in a cell is to be calculated. A formula record (type 16) is used for any cell with such a formula. Fortunately, the last calculated value of the formula (the number that is displayed) is also stored.

A formula record is, in effect, a double-precision value record plus a series of bytes containing a formula. With some minor modifications, type 16 records therefore can be processed in a manner that is similar to that used for type 14 records. A procedure to do this can be found in lines 1650 through 1780.

Lines 1660 through 1730 correspond to lines 1440 through 1510 in the procedure for double-precision values. Lines 1740 through 1770 take care of the formula information. The format, cell location, and last computed value require 13 bytes, so the formula is stored in the remaining (RECORDLENGTH%-13) bytes, which are simply read and ignored. If desired, this procedure could also be used for double-precision values, since type 14 records have a record length of 13, and the FOR loop in lines 1750-1770 will not execute in such cases.

THE RANGE RECORD

When a worksheet (or portion of one) is saved, 1-2-3 writes a record indicating the range from which the data came. The record (type 6) indicates the column and row at the upper left corner and the lower right corner of the range in which data were contained. Each value is stored as a two-byte integer.

For files saved with a /FS command, the "from" row and column are both zero, and the "to" row and colum indicate the lower right corner of the area containing data. For portions of files extracted with a /FX command, the values indicate the range that the area containing data occupied in the original worksheet. Since cell locations are converted to a "base" of 0,0 when a portion of a worksheet is extracted, the "from" values must be subtracted from the "to" values for consistency. The resulting locations are printed by lines 1290-1310.

THE END-OF-WORKSHEET RECORD

The last record in a valid worksheet file is of type 1 and has a length of zero. The procedure between lines 1800 and 1820 processes it.

The program to read and print the essential information from a worksheet file is now complete. Incorporating these routines in other programs is relatively simple, making it possible to do any type of processing with a worksheet file.

To create a new worksheet file, the process must be reversed. A valid header record must begin the new file, and a valid end-of-worksheet record must end it. Including a range record after the header and before the data records is desirable (but not necessary). If possible, data records should be stored in order (left-to-right within rows, top-to-bottom by rows).

OTHER RECORDS

Files saved with 1-2-3 typically include a great deal of additional information. To print the contents of the records containing such information, add these lines:

```
305 PRINT "Record Type ";
  RECORDTYPE%
335 PRINT ASC (BYTE$ );
345 PRINT
```

These statements will print the type of each such record, followed by the numeric values of the bytes in it.

When you visit your dealer and compare the Princeton IBM-compatible HX-12 side-by-side with the IBM color monitor, your eyes will see the difference.

The HX-12 gives you higher resolution and finer dot pitch (.31mm) than the IBM 5153's medium resolution (.43mm) for a cleaner, sharper image.

Compare our full range of colors and our crisp whites without red bleed. You'll also see a difference in our non-glare screen—a feature your eyes will really appreciate in a long work session.

The Princeton HX-12 comes with a cable that plugs directly into the IBM PC, ready to burst forth into 16 superb colors. All at a suggested retail price ($695) that's a pleasure for sore eyes and overworked budgets.

Apple IIe users: call us to learn how you, too, can now enjoy the visible superiority of the Princeton HX-12.

Ask your local dealer for a demonstration and let your eyes decide. Or call us at 800-221-1490 for more information and the name of your nearest dealer.

If you're ready to move up to color, graduate to the Princeton HX-12. It's right at the head of its class.

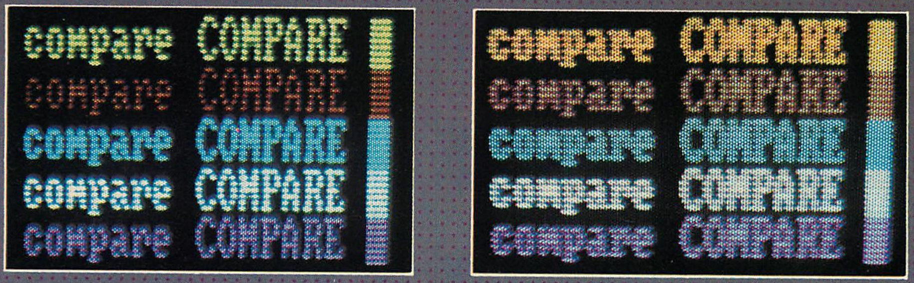
Now available at Computerland, Entré and local independent computer dealers.

Nationwide service provided by Bell and Howell Service Company and MAI Sorbus Service Company.

PGS Princeton Graphic Systems
an Intelligent Systems Company

1101-I State Road Princeton New Jersey 08540
(609 683-1660 TLX:6857009 PGS Prin.
800-221-1490

DON'T COMPROMISE:



OURS: .31 mm dot pitch, 80 column text. **THEIRS:** .43 mm dot pitch, 80 column text.

THE PRINCETON HX-12
HIGH RESOLUTION RGB COLOR MONITOR

NEW: adapter available for PCjr.

for the image your eyes deserve

Listing 1 INSIDE.BAS: A Program to Print Data in a Worksheet File

```
10   ' Program to print data in a Worksheet File
20   '       W. F. Sharpe, February 1984
30   '
40   ' Main Program
50        GOSUB 120 ' open file
60        RECORDTYPE% = 0
70        WHILE RECORDTYPE% <> 1
80           GOSUB 210 ' read and process a record
90        WEND
100    END
110  '
120  ' Procedure to open the worksheet file
130        INPUT "Worksheet name: "; FLNAME$
140        FLNAME$=FLNAME$+".WKS"
150        OPEN FLNAME$ AS 1 LEN=128
160            FIELD #1, 128 AS BUFFER$
170        GET #1
180        POINTER% = 1
190    RETURN
200  '
210  ' Procedure to read and process a record
220          GOSUB 370 ' read record type and length
230  '     go to appropriate routine for processing
240          IF RECORDTYPE% =  0 GOTO 1010 ' header
250          IF RECORDTYPE% =  6 GOTO 1120 ' range
260          IF RECORDTYPE% = 13 GOTO 1340 ' integer value
270          IF RECORDTYPE% = 14 GOTO 1430 ' double precision value
280          IF RECORDTYPE% = 15 GOTO 1540 ' character string (label)
290          IF RECORDTYPE% = 16 GOTO 1650 ' formula and value
300          IF RECORDTYPE% =  1 GOTO 1800 ' end of worksheet
310  '     not a type to be processed -- read remaining portion
320          FOR I%=1 TO RECORDLENGTH%
330               GOSUB 480 ' get the next byte
340          NEXT I%
350    RETURN
360  '
370  ' Procedure to get the type and length of the record
380  '    get record type
390            GOSUB 480
400            GOSUB 480
410            RECORDTYPE% = CVI ( PREVIOUSBYTE$ + BYTE$ )
420  '    get record length
430        .   GOSUB 480
440            GOSUB 480
450            RECORDLENGTH% = CVI ( PREVIOUSBYTE$ + BYTE$ )
460    RETURN
470  '
480  ' Procedure to get the next byte
490        PREVIOUSBYTE$ = BYTE$
500        BYTE$ = MID$(BUFFER$,POINTER%,1)
510        POINTER% = POINTER% + 1
520        IF (POINTER% <= 128) THEN RETURN
530           GET #1
540           POINTER% = 1
550    RETURN
560  '
570  ' Procedure to get format, row and column for a data record
580      GOSUB 480
590         FORMATBYTE$=BYTE$
600      GOSUB 480          .
610      GOSUB 480
620         COLUMN% = CVI ( PREVIOUSBYTE$ + BYTE$ )
630      GOSUB 480
640      GOSUB 480
650         ROW% = CVI ( PREVIOUSBYTE$ + BYTE$ )
660    RETURN
670  '
680  ' Procedure to print cell location
690  '   convert column to alphabetic characters
700       CHAR1% = COLUMN% \ 26
710       CHAR2% = COLUMN% MOD 26
720       IF CHAR1% = 0 THEN ALPHA$ = " " ELSE ALPHA$ = CHR$(64+CHAR1%)
730       ALPHA$ = ALPHA$ + CHR$(65+CHAR2%)
740  '   print column and row
```

```
750        PRINT ALPHA$; ROW%+1;
760    RETURN
770  '
780  ' Procedure to convert double precision number
790  '  test for NA code
800      IF ((BYT%(1)=255) AND (BYT%(2)=240)) THEN ISNA%=1 ELSE ISNA%=0
810      IF ISNA% = 1 THEN RETURN
820  '  test for zero
830      IF (BYT%(1)=0) AND (BYT%(2)=0) THEN DOUBLE#=0!: RETURN
840  '  get sign
850      IF ((BYT%(1) AND 128) >0) THEN SIGN%=-1 ELSE SIGN%=1
860  '  get exponent
870      BYT%(1) = BYT%(1) AND 127
880      BYT2LEFT% = (BYT%(2) AND 240)\16
890      BYT2RIGHT% = BYT%(2) AND 15
900      EXPONENT% = BYT%(1)*16 + BYT2LEFT% - 1023
910  '  get mantissa
920      SUM# = 0
930      FOR I% = 8 TO 3 STEP -1
940          SUM# = ( SUM# + BYT%(I%) ) / 256
950      NEXT I%
960      SIGNIFICAND# = 1 + (BYT2RIGHT%/16) + (SUM#/16)
970  '  compute value
980      DOUBLE# = SIGN% * (SIGNIFICAND# * (2^EXPONENT%))
990    RETURN
1000 '
1010 ' Procedure to process a header record (type 0)
1020     IF RECORDLENGTH% <> 2 THEN GOTO 1090
1030         GOSUB 480
1040             IF BYTE$ <> CHR$(4) THEN GOTO 1090
1050         GOSUB 480
1060             IF BYTE$ <> CHR$(4) THEN GOTO 1090
1070   RETURN
1080 ' error -- halt processing
1090     PRINT "ERROR -- Not a Valid Worksheet File"
1100     END
1110 '
1120 ' Procedure to process a range record (type 6)
1130 '  find range from which data were saved
1140         GOSUB 480
1150         GOSUB 480
1160             FROMCOL% = CVI ( PREVIOUSBYTE$ + BYTE$)
1170         GOSUB 480
1180         GOSUB 480
1190             FROMROW% = CVI ( PREVIOUSBYTE$ + BYTE$)
1200         GOSUB 480
1210         GOSUB 480
1220             TOCOL% = CVI ( PREVIOUSBYTE$ + BYTE$)
1230         GOSUB 480
1240         GOSUB 480
1250             TOROW% = CVI ( PREVIOUSBYTE$ + BYTE$)
1260 '  find lower right corner
1270         ROW% = TOROW% - FROMROW%
1280         COLUMN% = TOCOL% - FROMCOL%
1290     PRINT "Lower Right Corner: ";
1300     GOSUB 680 ' print cell location
1310     PRINT
1320   RETURN
1330 '
1340 ' Procedure to process an integer record (type 13)
1350     GOSUB 570 ' get format, row and column
1360     GOSUB 680 ' print cell location
1370       GOSUB 480
1380       GOSUB 480
1390           VALUE% = CVI ( PREVIOUSBYTE$ + BYTE$ )
1400     PRINT TAB(9); VALUE%
1410   RETURN
1420 '
1430 ' Procedure to process a double precision value record (type 14)
1440     GOSUB 570 ' get format, row and column
1450     GOSUB 680 ' print cell location
1460     FOR I% = 1 TO 8
1470         GOSUB 480
1480         BYT%(9-I%) = ASC ( BYTE$ )
1490     NEXT I%
1500     GOSUB 780 ' convert to double-precision number
1510     IF ISNA% = 1 THEN PRINT TAB(9); "NA" ELSE PRINT TAB(9); DOUBLE#
1520   RETURN
```

Hard Disk, Soft Shell

The Complete Hard Disk Solutions
For Desktop & Portable PC's 10 Megabytes $888, Or 20 Meg $1288.

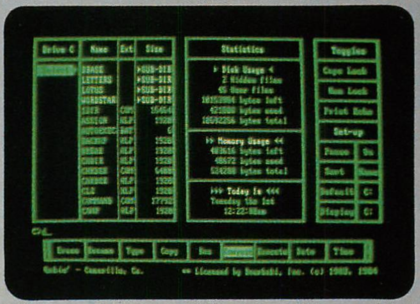
We have the whole package you need to add a hard disk to your IBM PC or compatible. Our half-height 10 and 20 megabyte drives have the lowest current draw in the industry. Unlike most aftermarket hard disks ours can "boot" direct from the hard disk. No software patches, or drivers to install just a copy of IBM PC DOS 2.0 or 2.1.

1dir fully Easy with the popular *1dir* menu driven "visual shell" software. It replaces complicated DOS commands with menus that allow you to just point at what you want to do. Help files explain DOS commands and give you on-line advice when you need it. *1dir* takes the wonder out of the tree structured subdirectories so useful in organizing a hard disk.

Hard Problems like excessive current draw and heat have until now been unsolved problems with aftermarket hard disks. Most drives draw lots of power. If your PC has many expansion boards in it, power to run a hard disk is probably not available. Hard disks have also been easily damaged by vibration and movement. And of course the problem any non-IBM product must face, compatibility with the IBM PC. We have tackled all these problems and come up with the best solutions available at any price.

Cool & In Control with half-height drives so efficient they *draw the same amount of power as a floppy disk drive*. This means a minimum of heat inside your PC with more power available for expansion boards. The controller uses LSI technology to provide fewer components, drawing less power, and giving significantly im proved performance over the IBM XT.

The Portable Plus is 10 or 20 megabytes to go. In fact, the drives we use have been selected by several computer makers for use in their portable computers. Their plated recording media helps withstand vibration, and allows the drive to be used at any angle.



Total Compatibility is a necessity. Our hard disks have the ability to boot directly from the hard disk. Check around, very few aftermarket systems can. In fact, all you need is a copy of DOS 2.0 or 2.1 and you are ready to go, no software drivers to install, or DEBUG patches to apply.

Why Buy From Us? It's simple really. Better value, and no hassles. You see, the price advertised is the whole price. No extra for freight, credit cards, COD fees, or insurance. Perhaps the best part is if you have a question you deal direct with Qubie'. We have the knowledge and ability to help you quickly. And, *if you do have a hardware problem during the warranty period it is fixed or replaced within 48 hours!*

No Risk. Don't be afraid to save. Our manual is written so even the novice can successfully install his own system. And remember, if for any reason you are not happy within 30 days of purchase you may return it for a full refund, and we'll pay the freight back! Get our competition to make the same offer, and find out which system is best. We know which one you will choose.

System Requirements: Any IBM PC with 64k RAM and PC DOS 2.0 or later. Compatible owners call for application information.

PC10: Includes 10 Megabyte drive, controller, cables, installation instructions, *1dir* software, and 1 year warranty. **$888**

PC20: Same as above with 20 Megabyte drive. **$1288.**

Options: Auxiliary power supply, for those with computers already full of power hungry expansion boards. #PCPWR $88.

Half-height bezel (specify computer), #HHBZL $15.

DEALER AND QUANTITY PURCHASE INQUIRIES INVITED

1dir Directory System Included

Order Today, Shipped Tomorrow!

Price includes UPS surface freight and insurance. Add $12 for two day air service. For fastest delivery send certified check or credit card. Personal checks take 18 days to clear. Calif. residents add 6% sales tax. Corporations and Institutions call for purchase order details.

(800) 821-4479
Toll Free Outside California

(805) 987-9741
Inside California

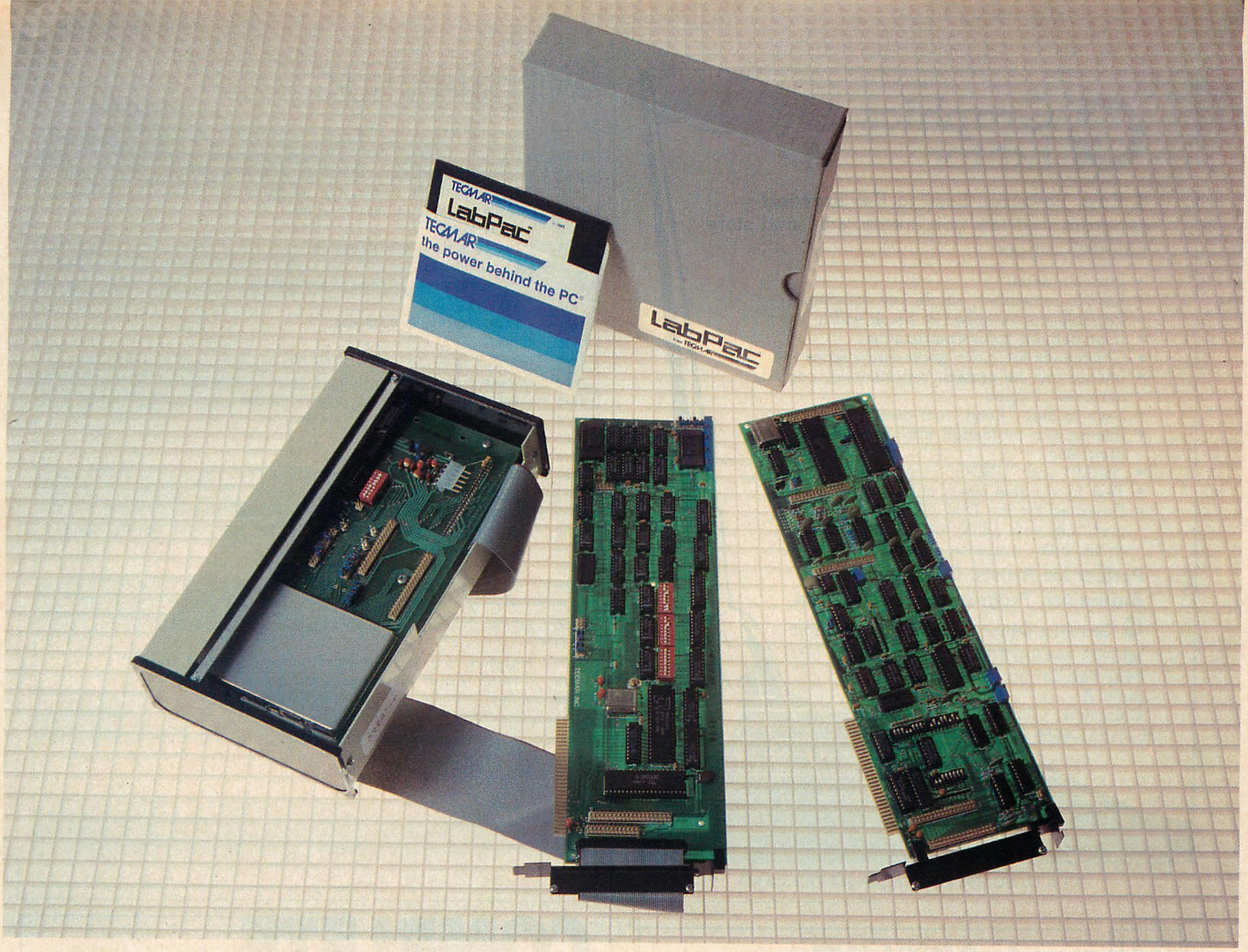
QUBIE'

4809 Calle Alto
Camarillo, CA 93010

Tempo House, 15 Falcon Road,
London SW11, United Kingdom

9/62 Blackshaw, Mortdale
2223 N.S.W., Australia

CIRCLE NO. 224 ON READER SERVICE CARD



Tecmar Provides Solutions

LabPac™ makes data acquisition quick and easy

Save your valuable time and money with LABPAC from Tecmar. Instead of spending days slaving over complex programs and languages, your data acquisition system can be up and running in hours. Now even novice programmers can easily design highly sophisticated data acquisition and process control systems. LABPAC provides a library of advanced assembly language subroutines that can be called by your BASIC, FORTRAN or Pascal programs. These subroutines can sample a channel, convert the signal, and store the value in memory for later use. LABPAC also contains many specialized commands for more complex procedures. In addition to all this, LABPAC also can operate in the background allowing you to sample, store, and analyze your data at the same time. LABPAC is compatible with the Tecmar Lab Master, Lab Tender and DADIO boards.

Lab Master™
End the tedium of monitoring your experiments or processes with the Lab Master. The Lab Master lets you save time and money by having your PC do the work for you. This board is powerful enough for the most demanding lab or industrial applications, yet with the LABPAC software support package it is easy to use. With Lab Master you have complete control over A/D conversion, D/A conversion, counting, timing, and digital I/O. This board is so advanced it can sample (with options) up to 256 channels of data and can convert a single channel up to 80,000 times in one second. The Lab Master doesn't sacrifice accuracy, since it comes with 12-bit precision, and has options for 14 and 16 bit accuracies. However, the most amazing feature of the Lab Master is that it can do all this and still cost less than $1,000.

Lab Tender™
The Lab Tender is the economical solution to your data acquisition needs. Whether you are on a tight budget or simply don't need all the precision of the Lab Master, the Lab Tender is the perfect board for you. The Lab Tender offers 32 channels of 8-bit A/D and 16 channels of 8-bit D/A, at a conversion rate of 50,000 samples per second. It also has counting, timing and digital I/O capabilities. No one else can offer you this combination of features for less than $500.

DADIO™
The DADIO extends the capabilities of the Lab Master or Lab Tender boards by providing 4 additional D/A channels and 3 I/O ports.

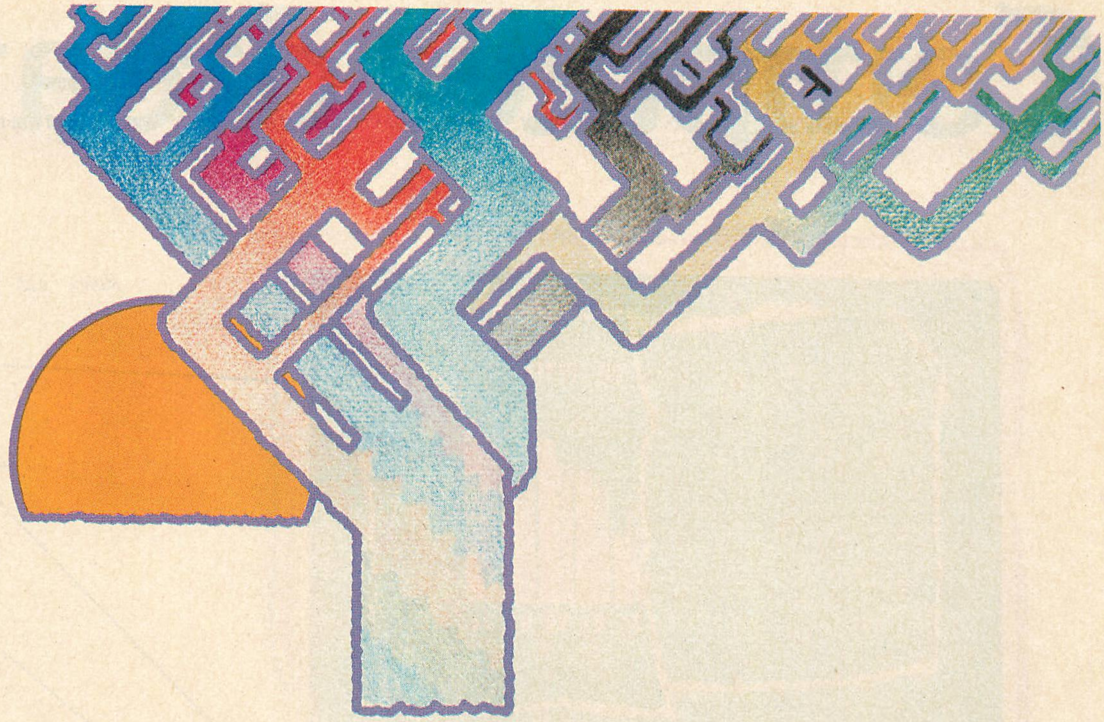
TECMAR
the power behind the PC™

Tecmar, Inc. 6225 Cochran Road Solon (Cleveland), Ohio 44139-3377 Phone: (216)349-0600 Telex: 466692

CIRCLE NO. 209 ON READER SERVICE CARD

WORKSHEET FILES

```
1530 '
1540 ' Procedure to process a character string record (type 15)
1550     GOSUB 570 ' get format, row and column
1560     GOSUB 680 ' print cell location
1570     CHARSTRING$ = ""
1580     FOR I% = 1 TO (RECORDLENGTH% - 5)
1590         GOSUB 480
1600         CHARSTRING$ = CHARSTRING$ + BYTE$
1610     NEXT I%
1620     PRINT TAB(9); CHARSTRING$
1630   RETURN
1640 '
1650 ' Procedure to process a formula record (type 16)
1660     GOSUB 570 ' get format, row and column
1670     GOSUB 680 ' print cell location
1680     FOR I% = 1 TO 8
1690         GOSUB 480
1700         BYT%(9-I%) = ASC ( BYTE$ )
1710     NEXT I%
1720     GOSUB 780 ' convert to double-precision number
1730     IF ISNA% = 1 THEN PRINT TAB(9); "NA" ELSE PRINT TAB(9); DOUBLE#
1740 ' read past formula bytes
1750     FOR I% = 1 TO (RECORDLENGTH% - 13)
1760         GOSUB 480
1770     NEXT I%
1780   RETURN
1790 '
1800 ' Procedure to process an end-of-worksheet record (type 1)
1810     PRINT "End of Worksheet File"
1820   RETURN
```



Btrieve and Xtrieve. B-tree file access for all your programming languages. With the query tool your users demand.

Introducing a powerful data base combination for PC application developers. Btrieve™, the most sophisticated file access method for your IBM™ PC. In single user and network versions. And Xtrieve™, a new menu-driven query tool that gives you—and your users—fast access to information.

Btrieve: for professional programmers. Btrieve provides fast, flexible file management for all your application development. All your programming languages—BASIC, Pascal, Cobol, C. With multikey access to records. Automatic file recovery. Unlimited records per file. Duplicate, modifiable, and segmented keys. Variable cache buffer.

Better applications faster. Based on the b-tree file indexing system, Btrieve provides optimal file management and eliminates writing file management routines. So you can develop *better* applications *faster.*

Xtrieve: easy window interface. The ideal complement to Btrieve, Xtrieve is the non-programmer's interface. Xtrieve's full relational capabilities let users define a virtual table of data from multiple files. Then Xtrieve speeds them through query building with a series of easy-to-follow windows.

No command language. Xtrieve is completely menu-driven, so there's no need to memorize command language. Or special syntax. Everything you need is on the screen. You and your users can perform unlimited queries—and get answers fast.

Xtrieve features a full range of restriction criteria. Online help messages. Automatic maintenance of information order by multiple indexes. And interfaces to access information from Lotus 1-2-3™ and dBase II® files.

Network capabilities. Network versions of Btrieve and Xtrieve allow data sharing in the most popular PC networks — PCnet™, NetWare™, EtherSeries™, MultiLink™, and OmniNet™.

For more information or to order, call or write:

SoftCraft Inc.

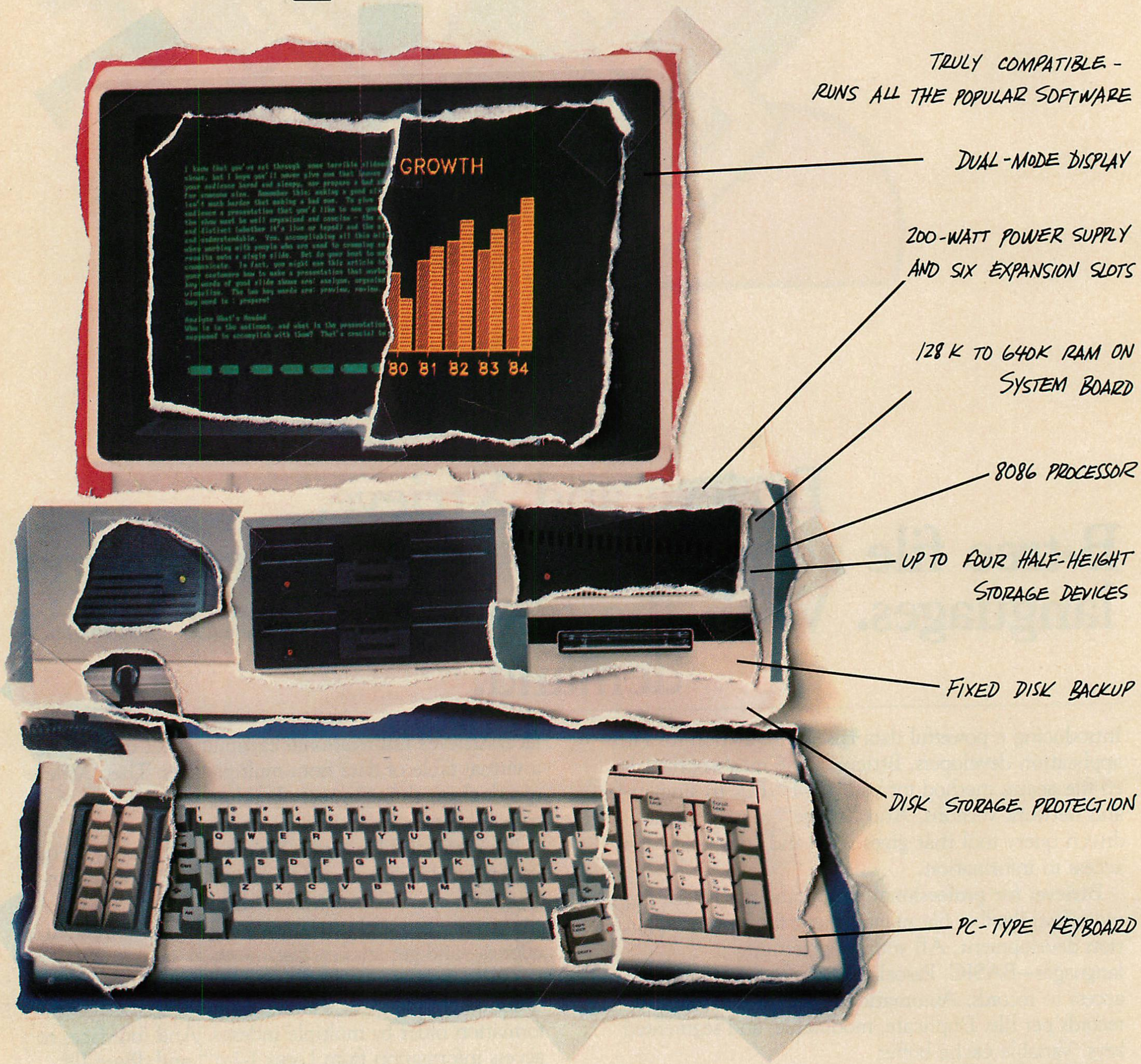
P. O. Box 9802 #590 Austin, Texas 78766 (512) 346-8380

Suggested retail prices: Btrieve, $245; Xtrieve, $195. Btrieve/N (network), $595; Xtrieve/N, $395. Dealer inquiries welcome. Btrieve requires PC-DOS or MS™-DOS 1.X or 2.X; Xtrieve, PC-DOS or MS-DOS 2.X.

Btrieve and Xtrieve, IBM, 1-2-3, dBase II, PCnet, NetWare, EtherSeries, MultiLink, OmniNet, and MS are trademarks of SoftCraft Inc., International Business Machines, Lotus Development Corp., Ashton-Tate, Orchid Technology, Novell Data Systems, 3Com Corp., Davong Systems Inc., Corvus Systems, and MicroSoft Inc.

CIRCLE NO. 201 ON READER SERVICE CARD

Computer dreams.



Introducing the new COMPAQ DESKPRO.™ COMPAQ combined the best features of the best desktop computers. And then added exclusives you won't find on others. Today they're together in *one* personal computer. The COMPAQ DESKPRO.

The COMPAQ DESKPRO is the most expandable desktop you can buy. It can grow as you grow, with up to six available expansion slots and four storage devices, including a unique fixed disk drive backup—all *inside the computer instead of out*, saving your desk from clutter.

It's the most rugged desktop, too, because it's the only desktop with a shock-mounted mass storage compartment to help keep disk drives in alignment and help protect you from losing data, downtime, and dollars.

It's also the most compatible desktop, giving you true hardware and software compatibility with the industry's most popular personal computers. The COMPAQ DESKPRO runs thousands of programs written for the IBM® PC and *XT* right off the shelf. Which brings us to the next point.

Dream computer.

The industry leader in compatibility.

Text and graphics on one screen, not two.

Power and space for almost unlimited expansion.

Add up to 640K of memory on system board and save expansion slots.

Runs software 2X to 3X faster.

10MB fixed disk drive
10MB fixed disk backup
Two diskette drives

Exclusive tape backup helps protect data.

Unique shock mounting helps protect your data and hardware investment.

Familiar layout enhanced with LED's on caps and numlock keys.

The COMPAQ DESKPRO is the fastest truly compatible desktop you can buy. It's two to three times faster than the industry standard. That saves time now. And in the future, as software becomes more sophisticated and integrated, it will save you even more time as well as protect your personal computer investment.

But there's more. We gave the COMPAQ DESKPRO one other feature not found on any other machine. You can choose between an amber or green dual-mode monitor that displays high-resolution text *and* graphics with equal brilliance, saving you the cost and clutter of a second monitor.

See the Dream Computer. Then pinch yourself. It's for real.

For the location of the nearest Authorized COMPAQ Computer Dealer, or for a free brochure, call 1-800-231-0900, Operator 5 or Telex #795437 COMPAQCOMP HOU.

The New

COMPAQ

DESKPRO™

It simply works better.

Make Stat Magic

Statistics, reports and plots happen magically with SPSS/PC™—*the* Statistical Package for IBM PC/XTs.*

SPSS/PC is the most comprehensive statistical package for performing simple or complex tasks, regardless of data size. It maintains feature and language compatibility with mainframe SPSS,® while optimizing for the PC environment.

Statistics range from simple descriptive to complex multivariate, including Multiple Regression, ANOVA, Factor and Cluster analysis. Loglinear and nonparametric procedures are also included.

Simple facilities allow transfer of files between SPSS/PC and programs like Lotus 1-2-3, dBase II and SAS. A complete Report Writer, Plotting facilities and a Communications program for mainframes round out a fully integrated product.

For more information, contact our Marketing Department without further ado. And see what a little stat magic can do for you.

SPSS Inc., 444 N. Michigan Avenue, Chicago, IL 60611, 312/329-2400.

In Europe: SPSS Benelux B.V., P.O. Box 115, 4200 AC Gorinchem, The Netherlands, Phone: +31183036711 TWX: 21019.

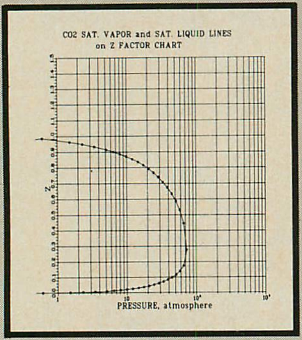
SPSS inc. PRODUCTIVITY RAISED TO THE HIGHEST POWER™

*SPSS/PC runs on the IBM PC/XT with 320K memory and a hard disk. An 8087 co-processor is recommended. Contact SPSS Inc. for other compatible computers.

IBM PC/XT is a trademark of International Business Machines Corporation. dBase II is a trademark of Ashton-Tate. 1-2-3 is a trademark of Lotus Development Corporation. SAS is a trademark of SAS Institute, Inc. SPSS and SPSS/PC are trademarks of SPSS Inc. for its proprietary computer software.

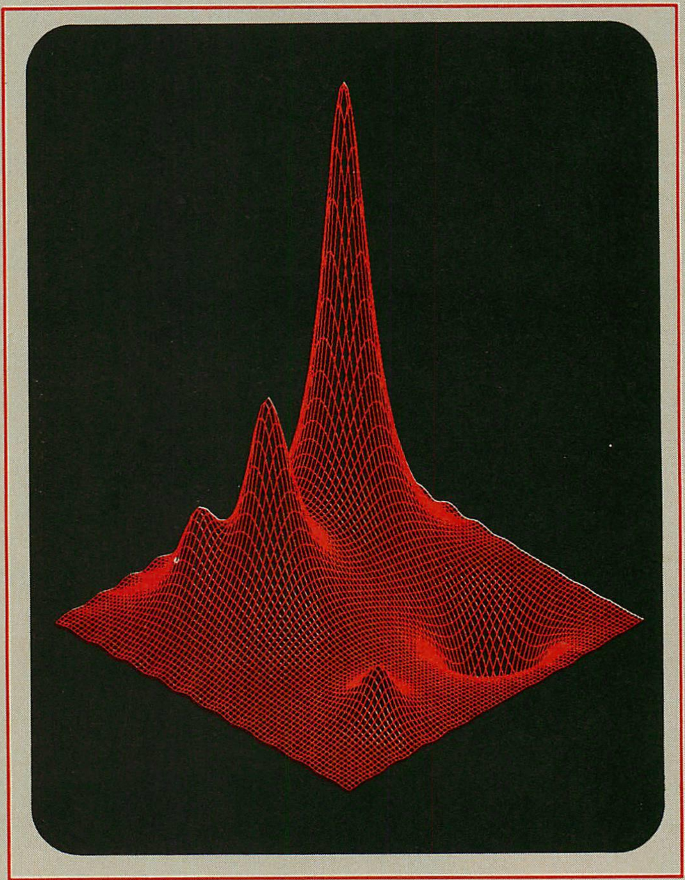
CIRCLE NO. 212 ON READER SERVICE CARD

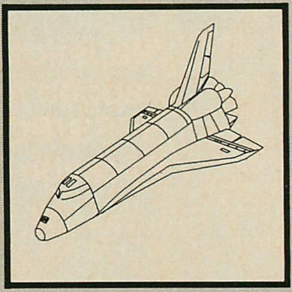
CONVERT YOUR PRINTER INTO A DIGITAL PLOTTER



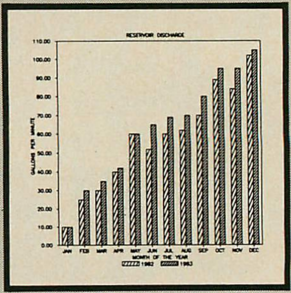
Scientific Plotting

Symbol Sets

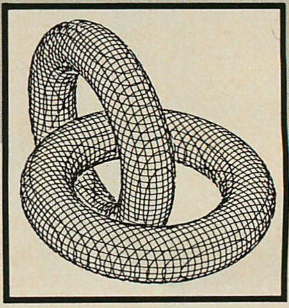




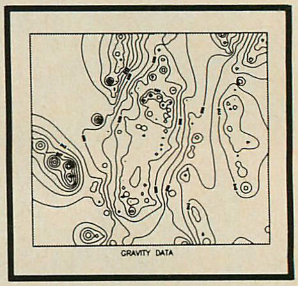
Digitizing



Business Graphics



Program Output



Contouring

PLOTCALL is a graphics system that processes standard plotter instructions into a form that can be printed on over 20 dot matrix printers. This is not another print screen program; presentation quality graphics are created directly on the printer to achieve the maximum resolution possible. (120 x 216 dots per inch for Epson printers)

PLOTCALL includes 17 symbol sets for sign making and plot labeling. Symbol sets may be altered, or create your own! Character strings may be created from any symbol set, rotated to any angle, and scaled to any size.

Plots can be created from any language or from digitized data. An interactive plot debugging program allows you to preview a plot on the screen before sending it to the printer.

All plots in this ad were created with **PLOTCALL** on an Epson printer. No additional hardware is required, not even the color graphics card!

Also included are the following easy to use, interactive programs that utilize the **PLOTCALL** system:

SURF creates high resolution three dimensional surface plots with hidden line removal.

TOPO creates contour maps using the same data as **SURF. TOPO** uses the character routines in the **PLOTCALL** system to provide inline contour labels.

GRAFIT creates charts and graphs of all types – XY graphs, bar charts, pie charts, and line graphs.

IBM PC or compatible with a minimum of 128K of memory and a dot matrix printer with graphics option are required.

ENTIRE PACKAGE ONLY **$199**
DEMO DISK (include printer model) **$10**

VISA AND MASTER CARD ACCEPTED

To order or obtain further information, contact:

CIRCLE NO. 184 ON READER SERVICE CARD

GOLDEN SOFTWARE, P.O. Box 281, Golden, CO 80402 – (303) 279-1021

Basic Elegance

RON BAUMAN

An annotated walkthrough of the program BASLDR that eases the process of calling subroutines from BASIC

Through a rich command set, Microsoft's BASIC interpreter allows even novice programmers not only to implement traditional programs, but also to access almost all of the graphics, sound, and interface capabilities of the IBM PC. Quite naturally, PC add-on manufacturers want to provide the same ease of access to their products.

Unfortunately, the mechanism provided by Microsoft to integrate third-party software into the BASIC environment is crude and nearly impossible for any but experienced software developers to understand. Loading callable subroutines for use by BASIC, even when understood, is a tedious and error-prone process, hardly suitable for end users accustomed to user-friendly applications.

But there is a more elegant way. Loading BASIC-callable subroutines can be made fast, easy and, most importantly, transparent to the end user. Three of PC-DOS's most powerful capabilities—the EXEC function call, environments, and dynamic memory allocation—can be used to load a callable machine language subroutine image, then load and execute the BASIC interpreter itself. From the user's point of view, invoking this program is just the same as invoking the BASIC interpreter. The program takes any of the command line arguments that are accepted by BASIC. The subroutines are loaded in memory without any intervention on the part of the user.

Ron Bauman is a consulting software engineer in the mini- and microcomputer industry. He is based in Ashland, MA.

Listing 1 is the assembly language program BASLDR, which loads the subroutine image and invokes BASIC on behalf of the user. BASLDR performs these tasks:

1. Releases its unneeded but allocated memory to DOS so there is room to load both the subroutine image and the BASIC interpreter;
2. Searches the current environment for an equivalence string that defines the location of the BASIC interpreter image to execute;
3. Determines the amount of memory that is needed to load the subroutine image and then allocates it via a DOS function call;
4. Loads the subroutine image via a DOS function call;
5. Stores the segment address of the loaded subroutine image where a BASIC program is able to read it;
6. Loads and executes the BASIC interpreter, passing the command line that is supplied by the user.

DOS passes control to BASLDR at location MAIN. There, the FAR return address of the PSP (Program Segment Prefix) is pushed on the stack. When the program finishes, it will perform a FAR RETurn, thus jumping to location 0 of the PSP, an INT 20H instruction (program terminate). The data segment is then established in DS.

At FILLBLK, the first steps are taken toward invoking the BASIC interpreter. The EXEC function call (4BH) requires a parameter block that describes the environment, command line, and default FCBs to be passed to the program to be invoked. Here the pointers to BASLDR's environment, command line, and FCBs are copied to the parameter block used to invoke the BASIC interpreter. The interpreter will inherit the same environment and command line arguments as BASLDR had on invocation. Appendix D of the *DOS Reference Manual* describes the EXEC function call.

DEALLMEM shows how to deallocate the memory not in BASLDR's code or data space. DOS's memory allocation mechanisms require some explanation.

DOS MEMORY ALLOCATION

When a program is invoked, either from the command prompt or by an EXEC function call, DOS loads the program and allocates all available memory to it (see figure 1). All programs should deallocate memory they do not need. Programs *must* deallocate memory if they intend to use either the memory allocation/deallocation DOS function calls or the EXEC function. DOS will only

allocate memory from its free memory pool. And it must allocate the memory required to load and execute a program.

The DOS function call 4AH (SETBLOCK) is used to modify the size of an allocated memory block. ES must contain the starting segment of the block and BX its new requested size in paragraphs (16-byte blocks). While not explicitly documented as such, the PSP segment of the program is the starting segment of the memory allocated to a program when it is loaded.

IBM, not well known for its software support, has some words of wisdom on setblock. To quote from its Customer Response #EF0014 and the DOS 2.1 *Technical Reference*, "A well-behaved program will use the SETBLOCK function to call when it receives control, to shrink its allocated memory block down to the size it really needs. . . " Many programs are not "well-behaved," including IBM's.

Unlike the way it closes all open files, DOS does not deallocate memory that a program has allocated when exiting. The program must perform explicit deallocation or the memory will be unavailable to subsequent programs.

Adhering to the rules of memory allocation/deallocation will be crucial in the long-awaited DOS 3.0 multitasking environment. Unfortunately, the current documentation fails to prepare the way.

Two versions of BASIC may be invoked, and, under DOS version 2.0, they can reside in subdirectories. Rather than having the user supply the location of the BASIC.COM file at each invocation of BASLDR or force it to reside in the current directory, BASLDR finds the location of BASIC in the "environment." This approach also eliminates the need to parse the BASLDR command line.

DOS reserves an area of low memory aligned on a paragraph boundary for storage of "environment strings." These strings, which are terminated with a byte of binary zeros, can be used to pass information to processes. In theory, environments look like an interprocess communication mechanism. The implementation has some problems, however. The segment address of a program's environment can be found at offset 2CH in its PSP.

At location FINDSTR, BASLDR looks for a string in its environment of the following form:

BASIC=dev:path\filespec

Dev:path\filespec is the fully qualified location of the BASIC interpreter that the user wishes to invoke. The user specifies this equivalence string before invoking the

BASLDR program with the following command:

SET BASIC=dev:path\filespec

This can be conveniently done in the AUTOEXEC.BAT file.

If the BASIC equivalence string cannot be found in the environment, BASLDR will try to locate BASICA.COM, then BASIC.COM in the current directory (location SDONE). The first of these that is found becomes the BASIC that will be loaded and executed.

The next step is to determine the amount of memory needed to load the subroutine image. Because the subroutine image is loaded as an overlay to BASLDR, the memory must already be allocated for it by BASLDR. At location READHDR, the .EXE file of the subroutines is opened and the file header read.

Appendix H of the DOS *Reference Manual* describes .EXE file structure and how a program is loaded. The size of the file in pages (512-byte blocks) is at offset 4 from the beginning of the header. While the size of the load module itself can be calculated exactly, it's easier to request a size equal to the file size in pages plus one. (The file size includes the header.)

EXEC FUNCTION CALL

At this point, the EXEC function call can be invoked to cause the subroutine image to be loaded as an overlay (see figure 1). The DOS EXEC function call (4BH) allows a program to invoke another program. This process, called *spawning*, is powerful" and quite easy.

The parent (invoking) process can specify the environment, command line, and two FCBs to the child (invoked) process by means of a parameter block. Once the EXEC function is called, DOS will allocate memory, load, and pass control to the child process just as if it had been invoked from the command prompt. When the child process exits, control is returned to the statement in the parent process following the EXEC function call.

The EXEC function call can also be used to load an overlay on behalf of a program. Again, a parameter block must be supplied to the function call. In this case, the segment location where the overlay is to be loaded and the segment fix-up value must be supplied in the parameter block. This simplifies using overlays, since the overlay's segment references are automatically adjusted by DOS to the values in the parameter block.

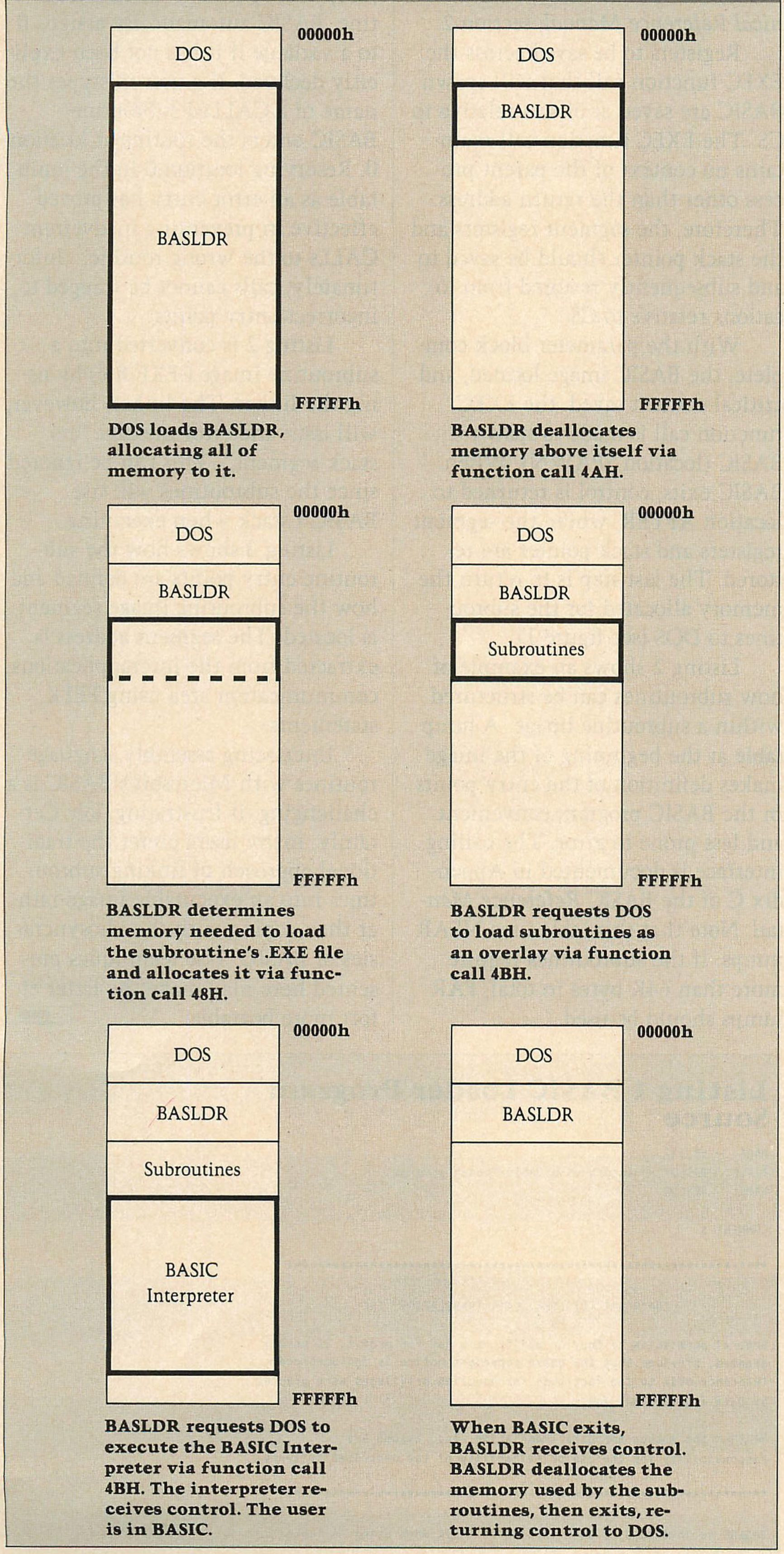
When a child process receives control, it does not receive a pointer to the parent's environment, but rather to a *copy* of the parent's environment. Thus, any changes that the child process might make to its environment will not be found by the parent process or by any other process spawned with the same environment address.

The environment must be contiguous. This causes some problems, even for DOS, which allocates only a small area of memory for the environment under version 2.0. It's quite easy to encounter an "out of environment space" while issuing SET commands if memory-resident routines have been loaded. The only recourse is to reboot, then issue the SET commands before loading any memory-resident routines.

In BASLDR at location LOADSUB, both the starting segment to load the subroutine image and the segment fix-up value are made the same as the starting segment of the memory block just allocated for the subroutine image. The EXEC function is then called to load the subroutine image file as an overlay.

For the user's BASIC program to call the subroutines, it will need to specify the segment where they reside via a DEF SEG statement. BASLDR places the segment address in the last two bytes of the "interapplication communications area" beginning at 0:4FE (location SAVSEG). This block of memory is mentioned

Figure 1: Memory Allocation during BALSDR Execution



only in the table entitled "Reserved Memory Locations" in the PC *Technical Reference Manual*, section 2.

Registers to be saved across the EXEC function call that will spawn BASIC are saved at offsets relative to CS. The EXEC function call maintains no context of the parent process other than the return address. Therefore, the segment registers and the stack pointer should be saved to and subsequently restored from locations relative to CS.

With the parameter block complete, the BASIC image located, and critical context saved, the EXEC function call is made to spawn BASIC (location SPAWN). When BASIC exits, control is returned to location AFTER, where the segment registers and stack pointer are restored. The last step is to return the memory allocated for the subroutines to DOS (see figure 1).

Listing 2 shows an example of how subroutines can be structured within a subroutine image. A jump table at the beginning of the image makes definition of the entry points in the BASIC program convenient and less prone to error. The calling interface is documented in Appendix C of the BASIC *Reference Manual*. Note that this table uses NEAR jumps. If the subroutines require more than 64K bytes in total, FAR jumps should be used.

The entry at offset 0 in the jump table points to an error routine. BASIC automatically assigns 0 to a variable if it has not been explicitly declared. If a user mistypes the name of a CALLed subroutine, BASIC enters the routine at location 0. Reserving location 0 in the jump table as an error entry has proved effective in preventing inadvertent CALLs to the wrong routine. Unfortunately, calls cannot be trapped to incorrect entry points.

Listing 2 is converted into a subroutine image (.EXE file) by using the linker. The linker, however, will issue the error message "no stack segment." This can be ignored since the subroutines will use BASIC's stack when executing.

Listing 3 shows how the subroutine entry points are defined and how the subroutine image segment is located. The segment address is extracted from the interapplications communication area using PEEK statements.

Interfacing assembly language routines with Microsoft's BASIC is a challenging, if frustrating, job. Certainly, many users prefer the traditional approach of linking subroutines into an executable image rather than coping with the idiosyncrasies of BASIC. The techniques presented here may make the latter effort more bearable.

Listing 1 BASIC Loader Program Source

```
PAGE    58,132
TITLE   BASLDR -- Interpreted BASIC loader program
NAME    BASLDR

COMMENT $

****************************************************************

            COPYRIGHT (C) 1984, DATA TRANSLATION, INC.

General permission to copy or modify, but not for profit, is hereby
granted, provided that the above copyright notice is included and
reference made to the fact that reproduction privileges were granted
by Data Translation, Inc.

Neither the author nor Data Translation, Inc. assume any
responsibility for the use or reliability of the described software.

****************************************************************

BASLDR is invoked by a user in lieu of BASIC when using callable
```

```
assembly language routines.  BASLDR performs the following functions:

- deallocates memory above itself for loading the routines and BASIC.

- scans the environment table for the SET string associated with BASIC=
  to find the location of the BASIC .COM image. If not found, attempts
  to use BASICA.COM in current directory. If not found, attempts to use
  BASIC.COM in the current directory.

- loads the callable subroutines image using the EXEC overlay DOS
  function.

- stores the segment of the routines in the interapplication
  communications area at 0:4FE, 0:4FF.  The BASIC user can determine
  the DEF SEG of the subroutines by doing a PEEK at these locations.

- spawns the BASIC specified above with the command line supplied by
  the user.

Note that any redirection of standard input and output is also passed
on to the the spawned BASIC.  This occurs transparently to BASLDR.

To create the executable image:

        MASM BASLDR,BASLDR,BASLDR;
        LINK BASLDR,,;

Written 3/19/84 by Ron Bauman

$
;
; Define the code, data and stack segments and groups.
;
PGROUP  GROUP   PROG
DGROUP  GROUP   DATA,STACK

PROG    SEGMENT BYTE PUBLIC 'CODE'
PROG    ENDS

DATA    SEGMENT WORD PUBLIC 'DATA'
;
_TOP    DW      0                       ;top of the stack address storage
;
COMMLOC EQU     4FEH                    ;interapplications comm area in low
                                        ;memory where segment of subroutine
                                        ;image is stored
BASFIL  DB      128 DUP (0)             ;storage for BASIC interpreter
                                        ;filespec
STRLOC  DW      0                       ;offset of BASIC filename string
SUBNAM  DB      'MYSUBS.EXE',0          ;name of callable subroutines image
;
ENVSTR  DB      'BASIC='                ;SET string in environment to be
                                        ;matched
ESTRLEN EQU     $ - ENVSTR              ;length of SET string for search
ADVBAS  DB      'BASICA.COM',0          ;advanced BASIC filespec
DSKBAS  DB      'BASIC.COM',0           ;disk BASIC filespec
;
HDRSIZ  EQU     32                      ;size of .exe file header to read in
                                        ;bytes
EXEHDR  DB      HDRSIZ DUP (0)          ;buffer to contain MYSUBS.EXE image
                                        ;header
IMGSIZ  EQU     4                       ;offset to .exe image size in .exe
                                        ;header
;
; PSP offset definitions
;
ENVIR   EQU     2CH                     ;segment address of passed
                                        ;environment
CMDLEN  EQU     80H                     ;size of string on command line
CMD     EQU     81H                     ;address of string passed on
                                        ;command line
FCB1    EQU     5CH                     ;first default FCB
FCB2    EQU     6CH                     ;second default FCB
;
; subroutine image loader parameter block
;
SBLOCK  DW      2 DUP (0)
SBSEG   EQU     0                       ; offset to segment address to load
                                        ; image
SBREL   EQU     2                       ; offset to word relocation factor
;
; BASIC image loader parameter block
;
BBLOCK  DW      7 DUP (0)
                                        ; define offsets in BBLOCK
BBENV   EQU     0                       ; segment address of environment
BBCMDP  EQU     2                       ; DWORD of command line to be passed
BBFCB1  EQU     6                       ; DWORD of first FCB
BBFCB2  EQU     10                      ; DWORD of second FCB
;
; error message strings
;
LEADR   DB 'BASLDR -- $'

NOTFMSG DB 'File MYSUBS.EXE was not found in the current directory.'
        DB 13,10,'$'
NORMMSG DB 'Not enough memory to load MYSUBS.EXE.',13,10,'$'
NOBASMSG DB 'BASIC program was not found.',13,10,'$'
;
DATA    ENDS

STACK   SEGMENT PARA STACK 'DATA'
SBASE   DW      8*16 DUP (?)            ;keep size a multiple of 16
STKEND  EQU     $                       ;end of stack
STACK   ENDS

PROG    SEGMENT BYTE PUBLIC 'CODE'      ;begin code
        ASSUME CS:PGROUP
;
;
MAIN    PROC    FAR
        PUSH    DS                      ;save the PSP address for return
        XOR     AX,AX                   ;and the offset
        PUSH    AX
        MOV     AX,DGROUP               ;set up DS register to address our
                                        ;data segment
        MOV     DS,AX
        ASSUME  DS:DGROUP
;
; fill the BASIC loader parameter block from our PSP while we have the
; PSP address in ES.
;
FILLBLK:
        MOV     BX,OFFSET BBLOCK        ;get parameter block address in BX
        MOV     AX,ES:ENVIR             ;load environment segment
        MOV     BBENV[BX],AX
        MOV     WORD PTR BBCMDP[BX],CMDLEN
                                        ;load the command line pointer
        MOV     BBCMDP+2[BX],ES
        MOV     WORD PTR BBFCB1[BX],FCB1
                                        ;load FCB1 pointer
        MOV     BBFCB1+2[BX],ES
        MOV     WORD PTR BBFCB2[BX],FCB2
                                        ;load FCB2 pointer
        MOV     BBFCB2+2[BX],ES
;
; deallocate memory above our stack for the subroutines and interpreter
;
DEALLMEM:
        MOV     AX,OFFSET STKEND        ;get the end of the stack
        MOV     _TOP,AX                 ;save it
        MOV     CL,4                    ;make it a paragraph value
        SHR     AX,CL
        MOV     BX,SS                   ;get the stack segment
        ADD     BX,AX                   ;end of program in paras.
        MOV     AX,ES                   ;start of PSP and memory to
                                        ;deallocate
        SUB     BX,AX                   ;final requested size of memory
        MOV     AH,4AH                  ;SETBLOCK DOS function request
        INT     21H
;
;  extract the location of the BASIC program we are to invoke. First
;  we scan the environment for a string equated to "BASIC=".
;
FINDSTR:
        MOV     AX,BBLOCK+BBENV
                                        ;get the environment seg address in ES
        MOV     ES,AX
;
        XOR     DI,DI                   ;ES:DI points to start of environment
        CLD                             ;forward string comparison
;
SCNENV:
        CMP     BYTE PTR ES:[DI],0
                                        ;zero byte is end of environment
        JZ      SDONE
```

```
        MOV     SI,OFFSET ENVSTR
                                ;DS:SI points to string to find
        MOV     CX,ESTRLEN      ;length of comparison
        REPE    CMPSB           ;compare until a mismatch
        OR      CX,CX           ;did the SET string match completely?
        JZ      FOUND
        XOR     AL,AL           ;skip to end of this environment string
SAGAIN: SCASB
        JNE     SAGAIN
        JMP     SHORT SCNENV
;
;
; here for BASIC found. ES:DI points to BASIC file string. copy it to
; our area for the spawn call.
;
FOUND:
        MOV     SI,OFFSET BASFIL
SMOV:   MOV     AL,ES:[DI]
        MOV     [SI],AL
        INC     SI
        INC     DI
        OR      AL,AL
        JNZ     SMOV
;
        MOV     DX,OFFSET BASFIL
        MOV     AH,3DH          ;try to open it
        XOR     AL,AL           ;for read
        INT     21H
        JNC     BASFND
;
; here for no environment equivalence. Try to open the advanced then
; disk BASIC images.
;
SDONE:
        MOV     DX,OFFSET ADVBAS
                                ;name of BASIC image.
        XOR     AL,AL           ;request a read on open
        MOV     AH,3DH          ;open file DOS function
        INT     21H
        JNC     BASFND
;
        MOV     DX,OFFSET DSKBAS
                                ;name of BASIC image.
        XOR     AL,AL           ;request a read on open
        MOV     AH,3DH          ;open file DOS function
        INT     21H
        JNC     BASFND
;
        JMP     NOBASIC         ;error, goto file not found.
;
BASFND:
        MOV     STRLOC,DX       ;save offset to file string for later
                                ;invocation
        MOV     BX,AX
        MOV     AH,3EH          ;close the BASIC we have opened.
        INT     21H
;
; open the subroutine image file to get it's size for allocation and
; loading
;
OPNSUB:
        MOV     DX,OFFSET SUBNAM
                                ;name of subroutine image.
        XOR     AL,AL           ;request a read on open
        MOV     AH,3DH          ;open file DOS function
        INT     21H
        JNC     READHDR
        JMP     FNOTFND         ;error, goto file not found.
;
; Read the header created by the linker to get the image size
;
READHDR:
        MOV     BX,AX           ;file handle to BX
        MOV     CX,HDRSIZ       ;size of header area
        MOV     DX,OFFSET EXEHDR
                                ;address of buffer to hold header
        MOV     AH,3FH          ;read the header.
        INT     21H
;
; close the file
;
        MOV     AH,3EH
        INT     21H
```

```
;
; get the image size (in 512-byte pages) and convert to paragraphs
;
CALCSIZ:
        MOV     BX,WORD PTR EXEHDR+IMGSIZ
        INC     BX              ;add extra page to ensure remainder
                                ;will fit
        MOV     CL,5
        SAL     BX,CL           ;mulitiply by 32
;
; request that much memory for loading the subroutine image and invoke
; the load.
;
        MOV     AH,48H          ;allocate memory function
        INT     21H
        JNC     LOADSUB
        JMP     NOROOM          ;memory allocation failure
LOADSUB:
        MOV     SBLOCK+SBSEG,AX ;segment load address of subroutine
                                ;image
        MOV     SBLOCK+SBREL,AX ;also the segment fixup value
;
        MOV     DX,OFFSET SUBNAM
        MOV     AX,DS
        MOV     ES,AX           ;set up pointer to param block
        ASSUME  ES:DGROUP
        MOV     BX,OFFSET SBLOCK
        MOV     AL,3            ;overlay load
        MOV     AH,4BH          ;EXEC function call
        INT     21H
;
; store the segment of the image in the interapplication
; communications area
;
SAVSEG:
        XOR     AX,AX           ;segment 0
        MOV     ES,AX
        MOV     AX,SBLOCK+SBSEG ;get the segment where the subs were
                                ;loaded
        MOV     ES:COMMLOC,AX   ;save it in the area
;
; set up the parameters for the EXEC function call.
;
BPSET:
        MOV     AX,DS           ; get address of param block in ES:BX
        MOV     ES,AX
        ASSUME  ES:DGROUP
        MOV     BX,OFFSET BBLOCK
        MOV     DX,STRLOC
                                ; get address of file to execute
                                ; in DS:DX
;
; ready to spawn the interpreter. Save the necessary registers.
;
SPAWN:
        MOV     CS:SSSAV,SS
        MOV     CS:SPSAV,SP
        MOV     CS:BPSAV,BP
        MOV     CS:DSSAV,DS
;
        MOV     AL,0                    ;load and execute function
        MOV     AH,4BH                  ;EXEC function
        INT     21H
;
AFTER:
        MOV     BX,CS:SSSAV             ;RESTORE REGS
        MOV     SS,BX
        MOV     BX,CS:SPSAV
        MOV     SP,BX
        MOV     BX,CS:BPSAV
        MOV     BP,BX
        MOV     BX,CS:DSSAV
        MOV     DS,BX
;
; deallocate the memory used for the subroutines
;
RETMEM:
        MOV     AX,SBLOCK+SBSEG         ;get segment from param block
        MOV     ES,AX
        MOV     AH,49H                  ;FREE ALLOCATED MEMORY function
        INT     21H
;
        RET                             ; all done
```

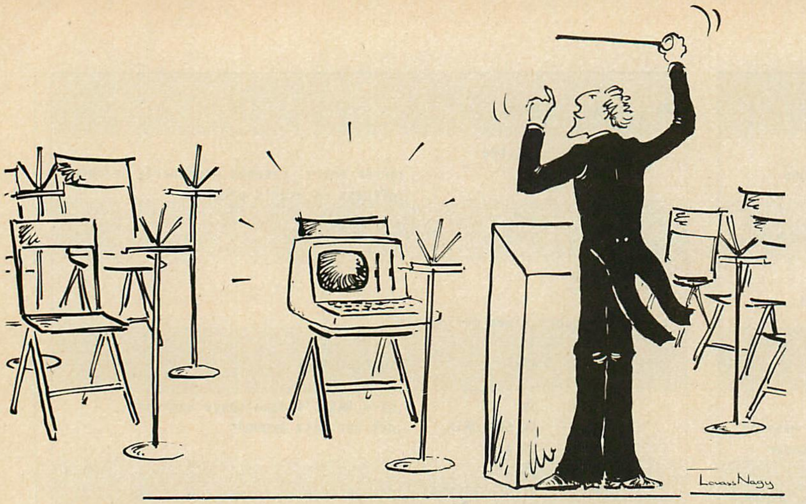
```
;
; error handling
;
FNOTFND:
        MOV     DX,OFFSET NOTFMSG       ;subroutine image not found
        JMP     SHORT PRTERR
NOROOM:
        MOV     DX,OFFSET NORMMSG       ;not enough memory to load subs
        JMP     SHORT PRTERR
NOBASIC:
        MOV     DX,OFFSET NOBASMSG      ;BASIC filespec not found
        JMP     SHORT PRTERR
;
; print the error message
;
PRTERR:
        PUSH    DX                      ;save the specific error message
        MOV     DX,OFFSET LEADR         ;get the error message leader
        MOV     AH,9H                   ;print string function
        INT     21H
        POP     DX                      ;print specific message after it
        MOV     AH,9H                   ;print string function
        INT     21H
;
        MOV     AX,_TOP                 ;reset stack for return call
        SUB     AX,4                    ;point to long return via PSP
        MOV     SP,AX
        RET
MAIN    ENDP
;
; temporary storage of registers across EXEC call
;
REGSAV  EQU     $

SSSAV   DW      0
SPSAV   DW      0
BPSAV   DW      0
DSSAV   DW      0
;
PROG    ENDS
        END     MAIN
```

Listing 2 Subroutine Image Example Source

```
PAGE 58,132
TITLE MYSUBS -- example BASIC-callable subroutine image
NAME MYSUBS
COMMENT *

This program may be freely copied and distributed.  The author
assumes no responsibility for its use or reliability.
This program defines the dispatch table and entry points for
BASIC-callable subroutines.  The subroutine image is created
with the following commands:

        MASM MYSUBS,MYSUBS,MYSUBS;
        LINK MYSUBS,,;

Written 4/15/84 by Ron Bauman

*
;
CSEG    SEGMENT BYTE PUBLIC 'CODE'
        ASSUME  CS:CSEG
;
DATA    SEGMENT WORD PUBLIC 'DATA'
;
;  define message strings
;
ERRMSG  DB '    Invalid subroutine call.',13,10,'$'
MSG1    DB '    Subroutine 1 executed.',13,10,'$'
MSG2    DB '    Subroutine 2 executed.',13,10,'$'
;
DATA    ENDS
;
; Define the dispatch table.  The order of entries in this table must
; correspond exactly with the order of name offset definitions in the
; user's BASIC program.  Note that it must appear at location 0 in the
; subroutine image.  This also means its segment must be defined first
; to the linker.
;
MAIN    PROC    NEAR
;
        JMP     CALLERR
                                ;error entry. catches improperly defined
                                ;offsets in user's program.
        JMP     SUB1
        JMP     SUB2
;
MAIN    ENDP
;
; CALLable subroutines definition
;
SUB1    PROC    FAR             ;subroutine 1
;
        PUSH    DS              ;save BASIC's data space address
        MOV     AX,SEG DATA     ;get our data segment
        MOV     DS,AX
        ASSUME  DS:DATA
        MOV     DX,OFFSET MSG1  ;set up pointer to message 1
        MOV     AH,9H           ;print the string to standard output
        INT     21H
        POP     DS
        RET
SUB1    ENDP
;
SUB2    PROC    FAR             ;subroutine 2
;
        PUSH    DS              ;save BASIC's data space address
        MOV     AX,SEG DATA     ;get our data segment
        MOV     DS,AX
        ASSUME  DS:DATA
        MOV     DX,OFFSET MSG2  ;set up pointer to message 2
        MOV     AH,9H           ;print the string to standard output
        INT     21H
        POP     DS
        RET
SUB2    ENDP
;
; The following routine catches calls to this image with an incorrectly
; assigned offset.  The usual case is that the user has mistyped the
; subroutine name, BASIC thought it was a new variable and assigned
; zero to it.
;
CALLERR PROC    FAR             ;subroutine 1
;
        PUSH    DS              ;save BASIC's data space address
        MOV     AX,SEG DATA     ;get our data segment
        MOV     DS,AX
        ASSUME  DS:DATA
        MOV     DX,OFFSET ERRMSG
                                ;set up pointer to our message
        MOV     AH,9H           ;print the string to standard output
        INT     21H
        POP     DS
        RET
CALLERR ENDP
;
CSEG    ENDS
        END
```

Listing 3 BASIC Program to Call the Subroutines

```
5  REM   PROGRAM MYPROG.BAS
10 REM   ---  BASIC example program for BASLDR  ---
15 REM   This program may be freely distributed and copied.
20 REM
25 REM   This program demonstrates how a user program
30 REM   accesses subroutines loaded via the BASLDR program.
35 REM
40 REM   Written 4/15/84 by Ron Bauman
50 REM
100 OFFSET%=3
110 SUB1=OFFSET% : OFFSET%=OFFSET%+3
120 SUB2=OFFSET% : OFFSET%=OFFSET%+3
200 DEF SEG=&H0  : REM get the subroutine image segment
210 VLSEG = PEEK ( &H4FE ) + 256*PEEK ( &H4FF )
220 DEF SEG=VLSEG  : REM set up subroutine image segment definition
230 CALL SUB1
240 CALL SUB2
250 END
```



Would you hire an entire band when all you need is one instrument? Of course not.

So why use a whole orchestra of computers when all you need is one to develop software for virtually any type of micro-processor?

The secret? Avocet's family of cross-assemblers. With Avocet cross-assemblers you can develop software for practically every kind of processor — *without having to switch to another development system along the way!*

Cross-Assemblers to Beat the Band!

Development Tools That Work

Avocet cross-assemblers are fast, reliable and user-proven in over 4 years of actual use. Ask NASA, IBM, Xerox or the hundreds of other organizations that use them. Every time you see a new micro-processor-based product, there's a good chance it was developed with Avocet cross-assemblers.

Avocet cross-assemblers are easy to use. They run on almost any personal computer and process assembly language for the most popular microprocessor families.

Avocet Cross-assembler	Target Microprocessor	CP/M-80	CP/M-86 IBM PC, MSDOS**
XASM04 *NEW*	6804	$ 250.00	$ 250.00
XASM05	6805	200.00	250.00
XASM09	6809	200.00	250.00
XASM18	1802/1805	200.00	250.00
XASM48	8048/8041	200.00	250.00
XASM51	8051	200.00	250.00
XASM65	6502/65C02	200.00	250.00
XASM68	6800/01, 6301	200.00	250.00
XASM75	NEC 7500	500.00	500.00
XASM85	8085	250.00	250.00
XASM400	COP400	300.00	300.00
XASMF8	F8/3870	300.00	300.00
XASMZ8	Z8	200.00	250.00
XASMZ80	Z80	250.00	250.00
XMAC682 *NEW*	68200	595.00	595.00
XMAC68K *NEW*	68000/68010	595.00	595.00

Your Computer Can Be A Complete Development System

Avocet has the tools you need to enter and assemble your soft-ware and finally cast it in EPROM:

VEDIT Text Editor makes source code entry a snap. Full-screen editing plus a TECO-like command mode for advanced tasks. Easy installation - INSTALL program supports over 40 terminals and personal computers. Customizable keyboard layout. CP/M-80, CP/M-86, MSDOS, PCDOS $150

EPROM Programmers let you program, verify, compare, read, display EPROMS but cost less because they communicate through your personal computer or terminal. No personality modules! On-board intelligence provides menu-based setup for 34 different EPROMS, EEPROMS and MPUs (40-pin devices require socket adaptors). Self-contained unit with internal power supply, RS-232 interface, Textool ZIF socket. Driver software (sold separately) gives you access to all programmer features through your computer, lets you download cross-assembler output files, copy EPROM to disk.

Model 7228 Advanced Programmer —Supports all PROM types listed. Super-fast "adaptive" programming algorithm programs 2764 in 1.1 minutes.

Model 7128 Standard Programmer — Lower-cost version of 7228. Supports all PROM types except "A" versions of 2764 and 27128. Standard programming algorithm programs 2764 in 6.8 minutes.

Model 7956 and 7956-SA Gang Programmers — Similar features to 7228, but program as many as 8 EPROMS at once. 7956-SA stand-alone version copies from a master EPROM. 7956 lab version has all features of stand-alone plus RS-232 interface.

EPROM: 2758, 2716, 2732, 2732A, 2764, 2764A, 27128, 27128A, 27256, 2508, 2516, 2532, 2564, 68764, 68766, 5133, 5143. **CMOS:** 27C16, 27C32, 27C64, MC6716. **EEPROM:** 5213, X2816A, 48016, I2816A, 5213H. **MPU (w/adaptor):** 8748, 8748H, 8749, 8749H, 8741, 8742, 8751, 8755.

7228	**Advanced Programmer**	**$ 549**
7128	**Standard Programmer**	**429**
7956	**Laboratory Gang Programmer**	**1099**
7956-SA	**Stand-Alone Gang Programmer**	**879**
PDV	**Driver Software**	**95**
481	**8748 Family Socket Adaptor**	**98**
511	**8751 Socket Adaptor**	**174**
755	**8755 Socket Adaptor**	**135**
CABLE	**RS-232 Cable (specify gender)**	**30**

HEXTRAN Universal HEX File Converter — Convert assembler output to other formats for downloading to development systems and target boards. Also useful for examining object file, changing load addresses, extracting parts of files. Converts to and from Intel, Motorola, MOS, RCA, Fairchild, Tektronix, TI, Binary and HEX/ASCII Dump formats. For CP/M, CP/M-86, MSDOS, PCDOS $250

Ask about UNIX.

AVOCET'S SUPERB 68000 CROSS-ASSEMBLER — With exhaustive field testing completed, our 68000 assembler is available for immediate shipment. XMAC68K supports Motorola standard assembly language for the 68000 and 68010. Macros, cross-reference, structured assembly statements, instruction optimization and more. Linker and librarian included. Comprehensive, well-written manual. XMAC682 for MK68200 has similar features.

Call us toll-free for some straight talk about development systems.

1-800-448-8500

(in the U.S. Except Alaska and Hawaii)

VISA and Mastercard accepted. All popular disc formats now available --please specify. Prices do not include shipping and handling -- call for exact quotes. OEM INQUIRIES INVITED.

*Trademark of Digital Research **Trademark of Microsoft

AVOCET SYSTEMS INC.™

DEPT.
804 SOUTH STATE STREET
DOVER, DELAWARE 19901
302-734-0151 TELEX 467210

THE WORLD'S MOST POPULAR UNIX SYSTEM FOR THE IBM PC AND LISA 2

Now, SCO brings the world's most popular version of AT&T's UNIX™ Operating System — and a comprehensive line of applications software — to the IBM® Personal Computer and the Apple® Lisa® 2.

XENIX™ from SCO is simply the best microprocessor-based UNIX you can buy. **No one else in the UNIX business provides you with better support, and no other version of UNIX gives you all these features:**

VISUAL SHELL: The XENIX visual shell provides you with a screen-oriented, menu-driven interface—with selectable color for the IBM PC.

MULTIUSER: With XENIX, 3 or more users may efficiently use a single IBM PC or Lisa 2 simultaneously.

MOST APPLICATIONS: With nearly 80% of all micro-based UNIX systems running XENIX, more applications are available for XENIX than for any other version of UNIX.

MULTISCREEN™: SCO's Multiscreen™ allows you to run multiple programs simultaneously and view each program's screen with a touch of a key.

MOST COMPLETE: XENIX is the most complete version of the UNIX System available for micros — including Microsoft's commercial enhancements and key extras from Berkeley 4.2 UNIX.

MICNET: Micnet local area networking doesn't require a systems programmer to set up or added hardware to run.

PC-DOS: XENIX lets you read and write PC-DOS files on the IBM PC—and a cross-development environment will be available soon.

DOCUMENTATION: SCO's XENIX documentation is written to meet the needs of the small systems marketplace — well-organized, clear, concise.

The powerful family of XENIX systems software and applications is available today from SCO — your UNIX Systems Software Company since 1978 — to set a new standard for personal computer productivity.

SCO
THE SANTA CRUZ OPERATION

YOUR UNIX SYSTEMS SOFTWARE COMPANY

500 CHESTNUT STREET
P.O. BOX 1900
SANTA CRUZ, CALIFORNIA 95061

408/425-7222

Let's go to work.

CIRCLE NO. 191 ON READER SERVICE CARD

BASIC String Manipulation

TECH NOTEBOOK 26

Undocumented techniques for working with strings on the IBM PC

DAN ROLLINS

In addition to the standard string-manipulation functions described in the BASIC manual, several other techniques are available.

Garbage Collection

Programs sometimes stall while BASIC reorganizes its limited string space. Garbage-collection time can be minimized by defining fewer strings—especially by dimensioning string arrays with a smaller number of elements. When a string array is no longer needed, use the ERASE command. BASIC will have fewer strings to reorganize, so the reorganization will go faster.

In a program that has a particularly critical period during which the garbage collector must not come into action, the FRE("$") command can be used to force garbage collection to occur.

Here is a trick that will minimize the number of times that garbage collection occurs: use the SWAP command to avoid assigning a temporary variable. For example, a string sort can be shortened dramatically by replacing

```
T$=A$(J) :A$(J)=A$(J+1)
  :A$(J+1)=T$
```

with

```
SWAP A$(J), A$(J+1)
```

Padding and Truncating

Programmers often write code to pad an input field with blanks. To create a 12-digit, right-justified numeric string from the value of the variable N, try using this:

```
N$=STR$(N)
  :N$=SPACE$(12-LEN(N$))+N$
```

Here's an alternative method:

```
N$=RIGHT$(SPACE$(12)+STR$
  (N),12)
```

This example creates a temporary string that is at least 13 characters long with leading spaces, then chops off all but the final 12 characters.

A similar technique can be used to pad an alphabetic string with trailing spaces. The following example will left-justify L.NAME$ in a field of NAME.LEN blanks:

```
L.NAME$=LEFT$(L.NAME$+
  SPACE$(NAME.LEN),
  NAME.LEN)
```

This technique also truncates L.NAME$ to the desired length if it happens to be too long.

Although the RIGHT$ function is quite useful, it can be awkward in some cases. For instance, to extract a sub-string that starts at position P in string A$, the following code could be used:

```
S$=RIGHT$(A$,LEN(A$)-P)
```

This causes an illegal function error, however, if P is greater than the length of A$. The MID$ function offers an elegant solution. After

```
S$=MID$(A$,P)
```

S$ will contain the characters starting from position P of A$. If A$ has fewer than P characters, then S$ will be equal to " " (the null string).

Upshifting

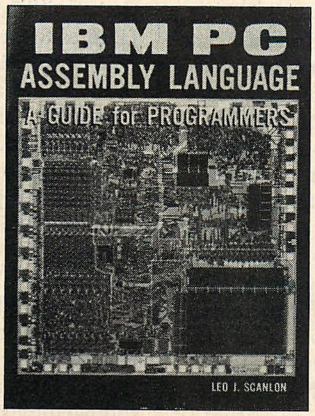
Only the most unfriendly programs ignore lower-case input. Unfortunately, BASIC has no intrinsic UPSHIFT$ function, but it is not difficult to build such a function into a subroutine. The following subroutine will force all characters in A$ into upper-case:

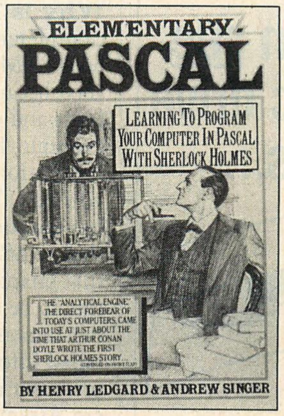
```
100 FOR J=1 TO LEN(A$)
110 C=ASC(MID$(A$,J,1))
120 IF C >= 97 AND C <= 122
  THEN MID$(A$,J,1)=
  CHR$(C-32)
130 NEXT :RETURN
```

This routine first converts the current character into a number, avoiding repeated calls to the BASIC string comparison routine (97 and 122 are the ASCII values of *a* and *z*, respectively, and 32 is the difference between *A* and *a*). This routine is most efficient when C and J have been declared as integers—the speed increase is quite noticeable in compiled BASIC programs. Notice, too, that no new string is created during the subroutine (the upshifted characters of A$ are placed back into A$ via the MID$= assignment statement), so garbage collection is never triggered.

These techniques will make programming in BASIC easier.

Dan Rollins is a software consultant and freelance writer who lives in California.





IBM PC Assembly Language: A Guide for Programmers
Leo J. Scanlon
(Robert J. Brady Co.; Bowie, MD; 1983)
311 pages, paper, $19.95

Although not every programmer wants or needs to program in assembly language, it is the only means of getting the computer under control for many small system owners. Assembly language provides faster programs, access to the powerful built-in BIOS routines, and access to ports—such as the game-adapter, parallel, and serial ports—from Pascal and APL.

Leo Scanlon's book has been on the market for more than a year. When it was published, it was the only real IBM assembly language book available; now there are half a dozen. A year ago I chose Scanlon's book as the main text for the Microprocessor and Assembly Language course at Broome Community College because it was the only book available. Next year, we will again be using Scanlon's book—this time because it is the best book available.

Complete and Detailed

Chapter 1 of Scanlon's text deals with the "whys" and "hows" of the 8088, and chapter 2 is a step-by-step guide to the use of the IBM Macro Assembler. In chapter 2, Scanlon shows the user how to edit, assemble, and run an assembly language program. He also includes a detailed example of the use of DEBUG for running programs and for examining register contents.

In chapter 3, details of the 8088 instruction set are covered. Instead of just dumping all of the 8088 commands in the user's lap, as some texts do, Scanlon presents a few commands or groups of related commands and gives an example of how they can be integrated into a program. Scanlon says, "This approach is intended to help you understand the instruction set, and how individual instructions "fit together," so you don't learn them as a bunch of disjointed entities." In contrast, David Bradley's book, *Assembly Language Programming for the IBM PC*, published by Prentice-Hall, handles the topic of the instruction set so poorly that it would be better to say that it isn't covered at all.

Topics in chapter 3 include: addressing modes (direct, indirect, relative, and indexed), instruction types, data transfer instructions (as a group), input/output instructions, flag instructions, arithmetic instructions, bit manipulation instructions, logical instructions, shift and rotate instructions, control transfer instructions, conditional transfer instructions (jumps, etc.), string instructions, move instructions for strings, and interrupt instructions. All of these groups have "little" example programs that teach the proper syntax and use for the instruction.

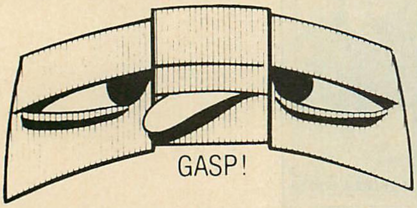
Chapter 4 is an introduction to high-precision mathematics. Because the 8088 and 8086 instruction set provides for multiplication and division, programmers are spared the necessity of writing programs to accomplish these tasks. In this chapter, Scanlon gives some detailed examples of 32-bit precision multiplication, division, and square roots. These examples teach both proper programming style and a more or less standard technique for achieving multiple-precision answers.

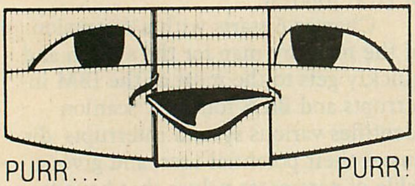
In chapter 5—"Data Structures"—Scanlon again teaches new concepts by using good examples. In assembly language, the concept of data structures usually refers to methods of handling things such as lists, tables, and files. Scanlon discusses how to enter and extract data from unordered and ordered lists, how to find maximum and minimum values in a list, and how to sort (bubble sort) a list. The subject of look-up tables (so often used to hold information on graphics characters) is presented as a fast approach to obtaining information on sines and cosines of angles as well as base (radix) conversions. Anyone interested in graphics work on the IBM PC should find this section of great interest.

Chapter 6 starts with a discussion of the memory map for the system and quickly gets to the meat of the IBM interrupts and BIOS routines. Scanlon identifies various system interrupts, discusses their potential uses, and gives a table of parameter values that have to be set when calling the interrupt. When dealing with the BIOS interrupts, he shows all of the necessary parameters that are so poorly detailed in the IBM *Technical Reference* manual. This section presents methods of using the BIOS routines to set the video mode, set cursor lines, set cursor position, read the cursor position, scroll a page, write a character to the screen, set color palette, put a dot on the screen (the beginning of graphics!), and more.

The subject of graphics is presented from the assembly language standpoint in chapter 7. It is somewhat disappointing that Scanlon limits his presentation to graphics that can be achieved on the monochrome screen (that is, IBM character set symbols that can be shown on the text screen). However, within the limits that are established, the presentation is handled well. Scanlon touches on the most important fundamental concepts of figure creation (using shape tables) and simple animation. This is an interesting introduction to a *fast* way to do graphics.

The final chapter is devoted to making sounds from the PC's built-in speaker. Scanlon gives an example called the "Speaker—Beeper Procedure" that will produce a tone of a specified frequency and duration. He then explains how to write a program that plays the song "Turkey In the Straw."





Why Is This Book So Good?

Completeness alone doesn't make a good book. *IBM PC Assembly Language* is good because it teaches important concepts and good programming style by example, its examples are simple and straightforward enough that a novice will be able to understand them, and it has questions at the end of each chapter that reinforce the concepts presented in that chapter. In addition, I have found this to be a good textbook for the beginner because of the first two chapters' detailed information on the 8088 microprocessor and how to use the IBM Macro Assembler.

There will always be some hard-core programmers who won't leave the first language they learned, whether it be BASIC, Pascal, FORTRAN, or something else. They say, "Why learn another language when I can do anything I want in BASIC or Pascal or FORTRAN?" My answer is this: If you program long enough and hard enough you will need the control of the computer that assembly language gives. When you do, *IBM PC Assembly Language* will be helpful. Scanlon's book is a great introduction to a topic that has received much bad press.

— WILLIAM H. MURRAY

Elementary Pascal: Learning to Program Your Computer in Pascal with Sherlock Holmes

Henry Ledgard and Andrew Singer
(Vintage Books, New York, 1982)
266 pages; paper, $12.95; hardcover, $19.50

This approach is certainly original, and the author (or one of them, at least) is no lightweight: Henry Ledgard is the author of the superb *Programming Proverbs* series. Does this unusual approach work? It all depends on what you assume the book ought to do.

The book is an exceedingly clever novel disguising a beginner's Pascal textbook. In the novel, Sherlock Holmes has obtained access to Charles Babbage's legendary Analytical Engine—which, the authors tell us, is mechanically microprogrammed (so we assume) to execute Pascal programs. Holmes greatly admires the machine's ability to work out logical problems and has begun using the Engine to solve cases that he used to solve by brain power alone. As he solves several mysteries with the Engine, he explains the art of programming to a skeptical Dr. Watson; the authors also provide concluding commentary after each Holmes lecture to clarify items that would be out of character for Holmes to explain.

I recognize Ledgard's organization of the material. He presents the *idea* of a program before ever displaying a single line of code. He establishes immediately his prime directive that understanding the problem comes *first*, by explaining the initial mystery and laying out the clues before starting in on the notion of writing a program to solve it.

The gradual explanation of the concepts and features of Pascal is done in exactly the right order. Program structure is explained first, followed by data, followed by statements and flow of control, followed by subprograms. The wrap-up is a lucid explanation of programming methods that is lifted essentially unchanged from Ledgard's *Programming Proverbs*.

The example programs each solve a Holmes mini-mystery. Although every bit as clever as the studied Victorian dialogue between Holmes and Watson, these programs are also every bit as contrived, and none of them provides insights into programming problems that occur in the real world, such as sorting files or battling nonstandard hardware.

So I return to the question of whether or not the book succeeds: Will it teach you Pascal? No. It operates at far too high a level of abstraction. To make the book universally applicable, Ledgard has chosen to ignore all real-world computers and implementations of Pascal and all the complications that come of using Pascal in the real world.

But will it teach you *about* Pascal? Yes, indeed. It explains clearly the methodology of top-down programming by using Pascal as an example language. Only the methodology is provided, however, not a full picture of the language.

In a way that's just as well; I can see Dr. Watson turning purple trying to understand Holmes's explanation of error trapping or DOS calls. The book is done with skill and a dry wit—anyone who enjoys Holmes will get a kick out of how well the authors have reproduced the mood of the original stories. As an entertainment that provides a review of structured programming it succeeds beautifully.

I recommend *Elementary Pascal* as a companion text to a more specific text, such as Cortesi and Cherry's *Personal Pascal*, which focuses on two real-life compilers on the IBM PC. Henry Ledgard's magic does not deal with Pascal so much as with top-down programming by successive refinement. A good book knows its limits. This one does.

— JEFF DUNTEMANN

MAX STUL OPPENHEIMER

Mom & Pop Hi-Tech, Inc.

Incorporating for all the wrong reasons

There are two ways to become president of a corporation. The first involves years of hard work, astute politics, and luck. This article is about the second.

With the exception of certain special classes of corporations, such as national banks, corporations are creatures of state law. The mechanics of forming a corporation are specified by statute and vary from state to state. The general pattern, however, is simple: a charter must be drafted and submitted to the appropriate state agency along with the appropriate filing fee. The charter is a purely utilitarian document, the literary style of which seldom rises above the level of "FIRST. The name of the Corporation is Mom & Pop Hi-Tech, Inc." The state filing fee is usually nominal.

Once the charter has been approved and filed, bylaws should be adopted, and stock should be issued. It is desirable to issue stock qualified as "1244 stock," which, under Section 1244 of the Internal Revenue Code, creates an ordinary rather than capital loss for its owner if it becomes worthless.

Returning to more optimistic thoughts, directors and officers should be elected, and then everyone should get back to work. In some states, certain formalities (such as the adoption of a corporate seal) must be attended to, and most banks will ask that corporate resolutions be adopted to open a bank account, (they usually have a preprinted standard form that can be used).

There are limits to how far this technique can go. For example, this method cannot be used to become President of the United States, because the state official charged with filing articles of incorporation would not accept the corporate name "The United States." Likewise, unless there has been a grievous lapse in communications in some remote corner of the country, the official would not accept "International Business Machines" as a corporate name. (Even if a corporation could be formed under that name, the name could not be used as a trademark in the computer field. Trademark rights are independent of corporate name-adoption rights although it is always a good idea to foreclose a competitor's use of a corporate name as well as a trademark.)

Once the first corporation has been successfully formed, it is mechanically simple to form the second. Assuming that no circumstances (including the law) have changed, all that is required is to change the section of the charter that specifies the corporate name and to pay another filing fee. With a PC, a printer, a primitive word processor, and a checkbook, corporations can be multiplied at will. Mergers, dissolutions, and hostile takeovers require fancier software; pinstripes are optional, but they help.

Those who are looking for a quick way to dress up their resumes and who plan never actually to operate the corporation do not need to read any further—except to note that a corporation is a separate legal entity, different from its incorporator, its stockholders, its officers, its directors, and its employees. That means that an Employer Identification number (the corporate equivalent of a Social Security number) is required, as are certain forms that keep the state advised of the corporation's continuing existence (these forms must be accompanied by the appropriate fee). In other words, there is a price to be paid. Because there is a price for incorporating (in terms of both initial fees and maintenance), the incorporator should have a reason to do so.

The reason most often given for incorporating is that a corporation provides a means to limit personal liability. If the corporation is not a sham (that is, if it has adequate capital and corporate formalities are observed), it is a separate legal entity, and, therefore, its obligations are not imposed on its stockholders, its officers, or its directors.

The corporate form does place limits on certain types of liability that a co-venturer could impose in a general partnership. General partners are deemed to be agents for one another within the scope of the partnership's business. Therefore, any general partner with "apparent

Max Stul Oppenheimer is a partner in the law firm of Venable, Baetjer, and Howard, located in Baltimore.

Serious Software For The C Programmer From Lifeboat.™

Lattice® C Compiler: *The serious software developer's first choice.*

Selected for use by IBM,® Texas Instruments, Wang,® MicroPro,® Ashton-Tate,™ IUS/Sorcim,® Microsoft® and Lotus™ to name a few of the many. Why? Lattice C is clearly the finest 16 bit C compiler available today.

—Renowned for speed and code quality.
—Fully compatible with the C standards set forth by Kernighan and Ritchie.
—Four memory model options offer you unsurpassed control and versatility.
—Superior quality documentation.
—Now includes automatic sensing and use of the 8087 chip.
—Widest selection of supporting add-on packages.

Halo™: *A graphics development package rapidly emerging as the industry standard.*

—140 graphics commands including plot, line, arc, box circle and ellipse primitives, bar and pie charts; pattern fill and dithering commands.
—New: multiple viewports and "stroke text" for angling, scaling and filling text.

C Food Smorgasbord™: *This beautifully written collection of C functions is a valuable time saver.*

—Library includes a binary coded decimal arithmetic package, level 0 I/O functions, a terminal independence package, IBM PC ROM BIOS access functions and much more.

Pmate™: *The premier editor for the programming professional.*

Pmate is a full screen editor with its own powerful macro command language:

—Perform on screen row and column arithmetic, alphabetize lists, translate code from one language to another, call up other macros.
—Customize Pmate almost any way you like.
—Contains 10 auxiliary buffers for storage of macros, text, subroutines.
—An "undo" feature allows the programmer to retrieve whole series of deleted items.

Additional C Tools Available From Lifeboat:

Panel™: Screen formatter and data entry aid.
Lattice Windows™: Windowing utility; create "Virtual Screens."
Plink-86™: The popular linker; includes extensive overlay capabilities.
Pfix86™: Dynamic debugging utility.
Pfix86 Plus™: Symbolic debugger with capacity to debug overlays.
Btrieve™: Database record access/retrieval library.
Phact: Multikeyed ISAM C-Function library.
Fabs: Fast access B-tree database function library.
Autosort: Fast sort/merge utility.
ES/P: 'C' program entry with automatic syntax checking and formatting.
Greenleaf Functions™: Library of over 200 popular C functions.
And much more.

YES! Please rush me the latest FREE Lifeboat™ catalog of C products.

*Company Name*__________ *Business Phone*__________

*Name*__________ *Title*__________

*Address*__________

*City*__________ *State*__________ *Zip*__________

Please check the category where Lifeboat can best help you:

☐ *Software development* ☐ *Corporate* ☐ *Education*
☐ *Dealer/distributor* ☐ *Government* ☐ *Other*__________

Call Direct: 1-800-847-7078 (In NY State: 212-860-0300)
Return coupon to: Lifeboat Associates™
1651 Third Avenue, New York, NY 10128.

TJ/10

CIRCLE NO. 139 ON READER SERVICE CARD

authority" (that is, who seems to be acting as one in this position normally would) can bind the entire partnership and all co-partners. An officer of a corporation can commit only corporate assets, not those of its stockholders. It is possible to establish limited liability in the partnership format by forming a so-called *limited partnership*. The limited partners are liable only to the extent of their investment in the partnership, but they lose this protection if they participate in the management of the limited partnership.

The notion of limited liability is, however, often illusory for a small corporation as well. Although the corporation's debts are not imposed on its stockholders, nothing prevents the stockholders from voluntarily assuming those debts. For example, a bank lending money to a fledgling corporation will usually ask that the stockholders voluntarily agree to guarantee their corporation's repayment of the loan—or the bank will not make the loan.

If the corporation's sole stockholder is also its only employee, the actions of the corporation may be difficult to separate from the stockholder—if the newly elected president of Mom & Pop Hi-Tech, Inc. drives the company van into a parked police car, can there be any doubt that both the stockholder and the corporation have a problem? Standard practice for corporations is to indemnify their officers and directors against liabilities arising from the performance of corporate duties, but an indemnity is merely a promise to reimburse; it is worth no more than the corporation has, and it does not relieve the officer or director of his liability to third parties. Remaining a sole proprietorship or partnership and buying adequate insurance is frequently less costly than incorporating.

A second reason given for incorporating is the ease of transferring shares of the company. This is true in many cases (assuming that the transfer is exempt from state and federal securities laws; registering with the SEC is not easy regardless of the form of the entity). If the goal is to give every stockholder equal rights in proportion to his holdings, the process simply involves authorizing, then issuing, the required number of shares.

The goal is often more complicated, however. A passive investor of capital may want greater security than the whiz-kid who will actually make the venture work; the whiz-kid may want a larger share of the upside potential. An owner may want to give his hard-working employees a stake in the business, but not let them have a role in its management. These goals can be

achieved (by creating more than one class of stock), but this is not a job for amateurs, and it is not necessarily easier to achieve using a corporate vehicle than it would be through a partnership agreement.

Another reason for incorporating is to centralize and simplify management of the company. For a sole owner, this makes no difference. If the company has several owners, the corporate form makes management easier for those stockholders who are also managers, but (without special provisions in the corporate charter or bylaws) it deprives nonmanagers of control and of access to information concerning the corporation's operations.

One more reason frequently posited for incorporating is the belief that incorporation will provide tax advantages. What incorporation actually provides is tax *differences*. (The IRS publishes a "Tax Guide for Small Business" [Publication 334], which is a good introduction to the subject—provided the reader keeps in mind that it was written by a tax collector. The publication is updated annually, and local IRS offices have copies available gratis.)

A sole proprietor simply runs his business, figures out how much he made or lost in the process, and, at tax time, fills out Schedule C. A partnership is more complicated, but is still essentially a conduit to the individual partners' tax returns.

A corporation (I repeat for the third and final time) is a separate entity. To get it running, assets must be transferred to it (which often can be accomplished tax-free). If the corporation produces a profit, it is taxed on that profit—unless its stockholders have elected so-called "S Corporation" treatment. The corporation must distribute its profits to stockholders who want their share, and the stockholders will be taxed on any dividends they receive. If the corporation produces a loss, it can use that loss to offset past or future profits; but unless S Corporation status has been elected, the company's stockholders cannot take advantage of the loss on their personal tax returns.

Another tax difference occurs in selling a company. Selling is simpler for a sole proprietor, but it is generally more costly as well. Assuming that the corporation is not deemed "collapsible," (a tax concept that is primarily meant to prevent incorporation for the sole purpose of converting inventory into capital assets, but that has a broader reach), the sale of stock will be a capital gain and may qualify for a reduced tax. When sole proprietorship assets are sold, they are taxed at the lower capital gains rate only if they are capital assets—inventory items treated as if they were sold in the ordinary course of business and taxed at ordinary income rates.

Whether or not these tax differences are tax advantages depends on individual circumstances. The decision to incorporate, and, after incorporation, to elect (and later terminate) S Corporation treatment, must be predicated not just on taxes, or centralizing management, or transferring shares, or limiting personal liability, but rather on the future of the enterprise—anticipated profits and losses and capital need. Try to find a lawyer or accountant with a spreadsheet program. Or, try to find one who has never heard of spreadsheets and offer to teach him or her how to use one in return for helping you decide whether to incorporate—you could become each other's best clients.

TECH RELEASES

Hardware, software, and other developments for the PC

PCjr Keyboard

HARDWARE

IBM has introduced new features designed to extend the power and versatility of the PC*jr*. These features include a new **keyboard,** an optional **IBM PC*jr* 128KB Memory Expansion Attachment,** an **IBM PC*jr* Power Expansion Attachment,** and the **IBM PC*jr* Speech Attachment.**

The new keyboard, now standard with all PC*jr*s, has 62 individually contoured, full-travel keys with a familiar and comfortable typewriter-like layout, appearance, and touch. It can communicate with the system unit via an infrared signal; an optional connecting cord is also available. The new keyboard will be available at no charge to current PC*jr* owners and to those who purchase a PC*jr* from existing supplies.

The IBM PC*jr* 128KB Memory Expansion Attachment adds 131,072 characters of user memory to a PC*jr*. Up to three of these attachments can be connected to the computer's side expansion port to boost a PC*jr*'s total memory to 512K. A program included with the memory attachment permits all or part of the expanded memory to be transformed into a so-called electronic diskette and used as if it were a second disk drive. Price: $325.

The IBM PC*jr* Power Expansion Attachment provides the extra power needed to connect more than one memory, speech, or printer attachment to a PC*jr*. The power attachment supports up to three side attachments in any combination. This attachment is required to use the IBM PC*jr* Cluster Attachment on PC*jr*s with an internal modem, diskette drive, or any side attachment. $150.

The IBM PC*jr* Speech Attachment is a speech synthesizer that contains 196 built-in words and sounds and that can record speech on a diskette through a microphone input. $300.

IBM
Entry Systems Division
P.O. Box 1328
Boca Raton, FL 33432
305-241-7614

CIRCLE 455 ON READER SERVICE CARD

C-Pack (Computer Packaging Technology, Inc.) has designed a new product to facilitate the operation of a network with as many as 16 dumb-terminal workstations. The **Overdrive** system comes configured either as one to three modules that plug directly into expansion slots or as a plug-in expansion chassis. It is available with memory in increments from 64K to one megabyte per user. A single-user system is $1,495; 4- to 16-user systems start at $4,995.

C-Pack
P.O. Box 129
Mohegan Lake, NY 10547
914-528-0011

CIRCLE 489 ON READER SERVICE CARD

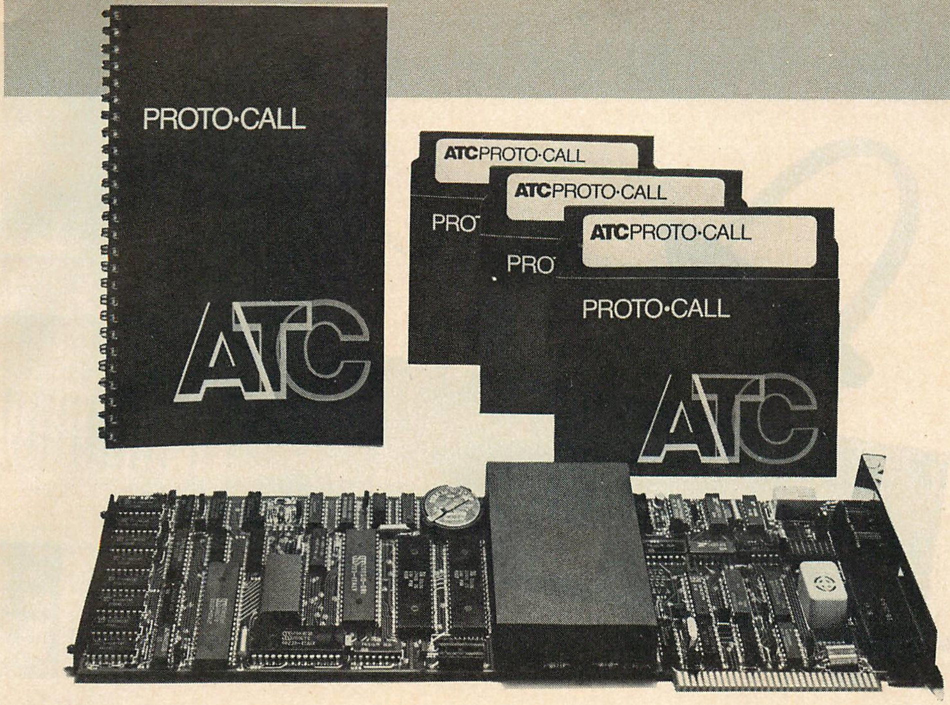
Overdrive

A product to protect PCs from power fluctuations is being offered by **General Interface Systems, Inc.** Called the **PC Uninterruptible Power Source,** the product provides up to 15 minutes of sine-wave power at a 400- or 650-watt output. It receives source power from the AC line and converts it to 48V DC, which it stores in quick-recovery gel-cell batteries. The unit will kick on whenever the commercial power source deviates from 110V AC. $850 for the 400-watt model; $1,495 for the 650-watt model.

General Interface Systems
12919 Alcosta Blvd.
San Ramon, CA 94583
415-838-2683

CIRCLE 487 ON READER SERVICE CARD

One of the first dual-function controller boards for Winchester disks and floppy tape has been introduced by **Sysgen, Inc.** The **SI500** controls two Winchester disks and two floppy disks in addition to one floppy tape drive. It offers full implementation of the ANSI-proposed Small Computer System Interface (SCSI). Sysgen has also developed the **SI520,** a multifunction controller for two Winchester disks and one 1/4-inch streaming tape drive. It supports the SCSI and QIC-02 interface. Both products have a 5 1/4-inch form factor and can support an SCSI data transfer rate of up to 1.25 megabytes per sec-

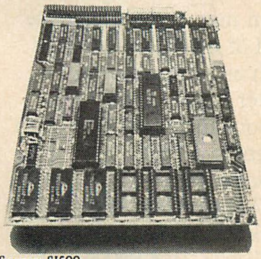


Proto.Call

ond. Prices: S1500, $400 in OEM quantities; S1520, $350 in OEM quantities.

Sysgen, Inc.
47853 Warm Springs Blvd.
Fremont, CA 94539
415-490-6770

CIRCLE 488 ON READER SERVICE CARD


Sysgen S1500

ATC Proto.Call, a communications package for the IBM PC and PC/XT, has been announced by **American Teleprocessing Corporation.** The product allows the user to run other software at the same time messages are being sent or received; to queue messages to be sent at other times; to send and receive while he is away from the machine; to have access to Telex, TWX, Dow Jones, NewsNet, and other databases and services; and to communicate with either touchtone or pulse dialing. The communications board plugs into a slot and is equipped with 64K of RAM, 103/212A modem, Z80 microprocessor, real-time clock with battery back-up, and Electronic Industries Association standard port. Proto.Call also comes with communications software. $995.

American Teleprocessing Corporation
0681 Haddington Drive
Houston, TX 77043
713-973-1616

CIRCLE 476 ON READER SERVICE CARD

Rapport Corporation has announced an expansion kit for the PC*jr* that turns the 64K, single-drive machine into a 512K, dual-drive Personal Computer. The **Drive Two Enhancement Package** allows the PC*jr* to run almost all software available for the IBM PC in addition to *jr* software. The package includes a 360K floppy disk drive, parallel printer port, clock calendar with battery back-up, and an expansion slot to increase RAM capacity to 512K. $675.

Also from Rapport comes the **Companion Keyboard Adapter Cable**, which will connect any IBM PC- or PC*jr*-compatible keyboard with the *jr*. The unit plugs directly into the computer and includes a six-foot coiled cord for the *jr* keyboard. Rapport also offers the **Companion Audio Amplifier,** a sound system for the PC*jr*. It includes a 1/4-watt amplifier as well as a 2 1/2-inch, 8-ohm speaker that allows complete audio reproduction when using any monitor at any sound level desired. Prices: Keyboard Adapter Cable, $39.95; Audio Amplifier, $24.95.

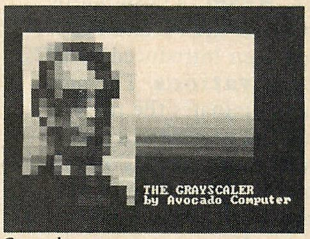
Rapport Corporation
80 S. Redwood Road
N. Salt Lake City, UT 84054
801-292-9454

CIRCLE 485 ON READER SERVICE CARD

Avocado Computer has developed a converter that upgrades any color graphics card to gray-scale operation, giving a green, white, or amber screen monitor the legibility, crispness, and power of an RGB display. Called the **Grayscaler,** the converter provides 16 levels of brightness and eliminates the "grain" caused by operating from the color card. $59.95.

Avocado Computer
17352 Yorkshire Avenue
Yorba Linda, CA 92686
714-528-1025

CIRCLE 480 ON READER SERVICE CARD

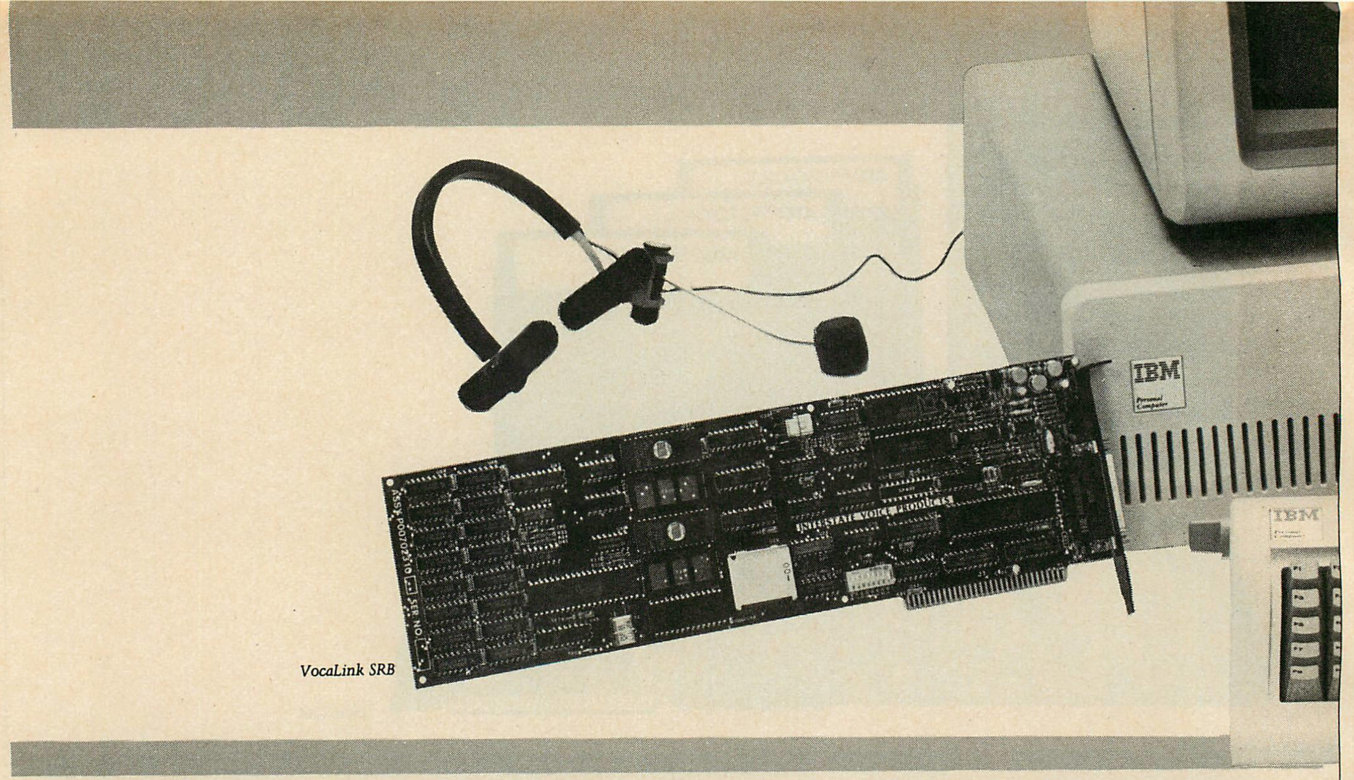


Grayscaler

Taurus Computer Products, Inc. has announced the **KS100,** an I/O peripheral system that combines an I/O board, termination panel, and software package to use in automated data acquisition and control applications. The product is available in two configurations: the KS101, which handles 16 analog input channels and 16 digital input/output points; and the KS102, which additionally offers 2 analog output channels and 4 event counters. A signal-termination panel provides screw terminal connectors in a configuration identical to the I/O board. Prices: KS101, $1,454; KS102, $1,857.

Taurus Computer Products, Inc.
340 Commercial Street
Manchester, NH 03101
603-623-7505

CIRCLE 483 ON READER SERVICE CARD



VocaLink SRB

PC Reset, a push-button reset for the PC and PC/XT, has been introduced by **PC Innovations.** The product eliminates the need to use the power switch as a reset button and allows the user to bypass the RAM check after the initial power is on. An optional remote-control assembly that can be located near the keyboard is available. PC Reset for the PC is $44.95; for the PC/XT, $49.95; and for the remote-control assembly, $29.95.

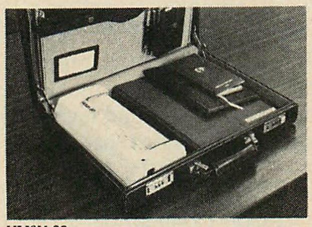
PC Innovations
P.O. Box 4374
Cherry Hill, NJ 08034
609-784-1133

CIRCLE 482 ON READER SERVICE CARD

A low-cost, dot matrix, thermal printer has been introduced by **Ergo Systems Inc.** The **HUSH 80** has 80-column bidirectional printing at 80 characters per second and graphics at 4,800 dots per square inch. Weighing in at 28 ounces, the printer is said to be the lightest portable printer available. It comes with an interface, cable, 100-foot roll of thermal paper, and 9-volt wall transformer with power cable. A nickel-cadmium battery pack is optional. The HUSH 80 is designed to fit into a conventional-size briefcase. $159.99.

Ergo Systems Inc.
1360 Willow Road
Menlo Park, CA 94025
415-322-ERGO

CIRCLE 478 ON READER SERVICE CARD


HUSH 80

A voice-recognition board that allows users to operate off-the-shelf software with up to 240 spoken commands has been developed by **Interstate Voice Products.** The discrete-word, speaker-dependent **VocaLink SRB** (Speech Recognition Board) occupies a single chassis slot and incorporates a high-speed, 16-bit Intel 80186 microprocessor. It supports two modes of operation: keystroke simulation, which substitutes user-defined verbal commands for keyboard entries to operate standard PC software, and PC mode for software written specifically to use SRB. Three microphone options are available: a headset with five-foot cord, headset with earphone, and desktop microphone operated with a foot switch. Features include a switch-selectable address to allow one PC to support four SRBs; a software microphone switch to allow the operator to activate or deactivate the SRB through spoken commands; a correction capability to cancel a spoken command by speaking a word defined by the user; and a message asking the user to repeat a word when the system cannot match the voice pattern to the vocabulary. $1,650.

Interstate Voice Products
1849 W. Sequoia Avenue
Orange, CA 92668
714-937-9010

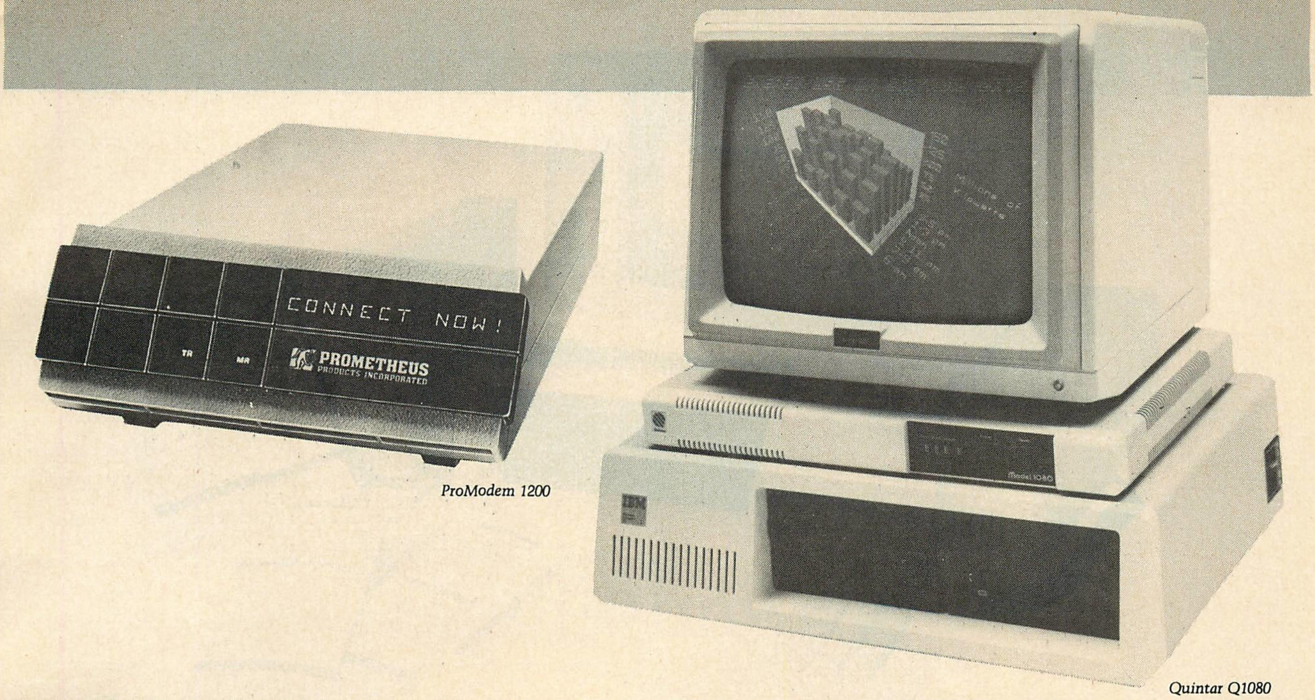
CIRCLE 486 ON READER SERVICE CARD

The PC*jr* now can support the Microsoft Mouse and some of the larger, more sophisticated software, such as Lotus 1-2-3, Microsoft Pascal and FORTRAN with the addition of **PC*jr* Booster with Mouse** from **Microsoft.** The hardware expansion card adds 128K RAM to *jr*, almost doubling performance speed on programs that require 128K or less. A battery-backed clock calendar and JBASIC, a software enhancement to the Microsoft BASIC cartridge sold for the *jr*, are included with PC*jr* Booster. JBASIC may double screen memory and increase speed of BASIC programs as much as 22 percent. PC*jr* Booster is $495 with 128K memory included; $295 with sockets for 128K.

Microsoft
10700 Northup Way
Box 97200
Bellevue, WA 98009
206-828-8080

CIRCLE 481 ON READER SERVICE CARD

An expansion multiplexer/amplifier system, the **EXP-16,** is now available from **Metrabyte.** The EXP-16 concentrates 16 differential analog input channels into one analog output channel and also provides signal amplification, filtering, and conditioning. The instrumentation amplifier provides user-selected gains of .5, 1, 2, 10, 50, 100, 200, and 1,000 plus



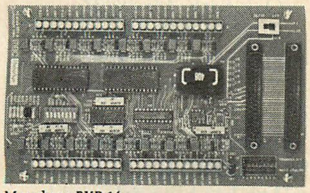
ProModem 1200

Quintar Q1080

user programming gain capability for special applications requiring specific gain. $345.

Metrabyte Corporation
254 Tosca Drive
Stoughton, MA 02072
617-344-1990

CIRCLE 484 ON READER SERVICE CARD



Metrabyte EXP-16

Quintar Corporation has developed a color graphics controller that enables the IBM PC to interface with a high-resolution color monitor, large-screen video projector, or slide-making device. The controller is offered in two versions: the **Q1000,** with resolution of 512 by 512 by 4; and **Q1080,** with resolution of 832 by 630 by 4. Both provide two full pages of display. Standard features include two 16- by 12-bit color look-up tables, each providing 16 simultaneously displayable colors from a palette of 4,096; four bits of programmable color information for each pixel; and the ability to combine alphanumerics and graphics on one screen. Q1000, $1,995; Q1080, $2,195.

Quintar Corporation
525 Maricopa Street
Torrance, CA 90503
213-320-5700

CIRCLE 477 ON READER SERVICE CARD

Prometheus Products has introduced a modem that is compatible with Hayes, Bell 103/212A modems. The **ProModem 1200** has a built-in clock/calendar, auto-answer and auto-dial, touch-tone and pulse dialing, built-in speaker with volume control, and optional buffer memory with up to 64K or storage. Also available is 12-character alphanumeric display to provide viewing of the operating status, diagnostic messages, phone numbers, and clock data. $495; optional buffer card, $99; optional alphanumeric display, $99.

Prometheus Products
45277 Fremont Blvd.
Fremont, CA 94538
415-490-2370

CIRCLE 475 ON READER SERVICE CARD

SOFTWARE

IBM has introduced the **IBM PC*jr* ColorPaint** cartridge program. This program requires a user-supplied mouse and provides the user with the ability to draw objects, use different brush styles and patterns, and "paint" with selections from a 16-color palette. $99.

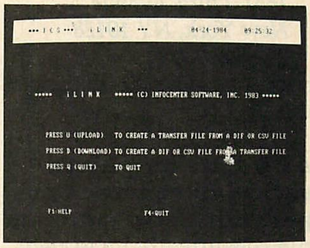
IBM
Entry Systems Division
P.O. Box 1328
Boca Raton, FL 33432
305-241-7614

CIRCLE 456 ON READER SERVICE CARD

InfoCenter Software, Inc. has announced **iLINK,** a dictionary-driven software transfer program that allows users to download, upload, or crossload data files from IBM PC database and spreadsheet applications to IBM mainframe information center products. iLINK is compatible with all PC applications programs that use DIF or CSV file formats for data management, graphics, and financial analysis (supported programs include Lotus 1-2-3, VisiCalc, SuperCalc, and dBASE II). Features of the package include an electronic note pad, partial source file extraction, record ID creation, field name and width modification, row/column transposition for each transfer procedure, and automatic default options. Price is $12,500.

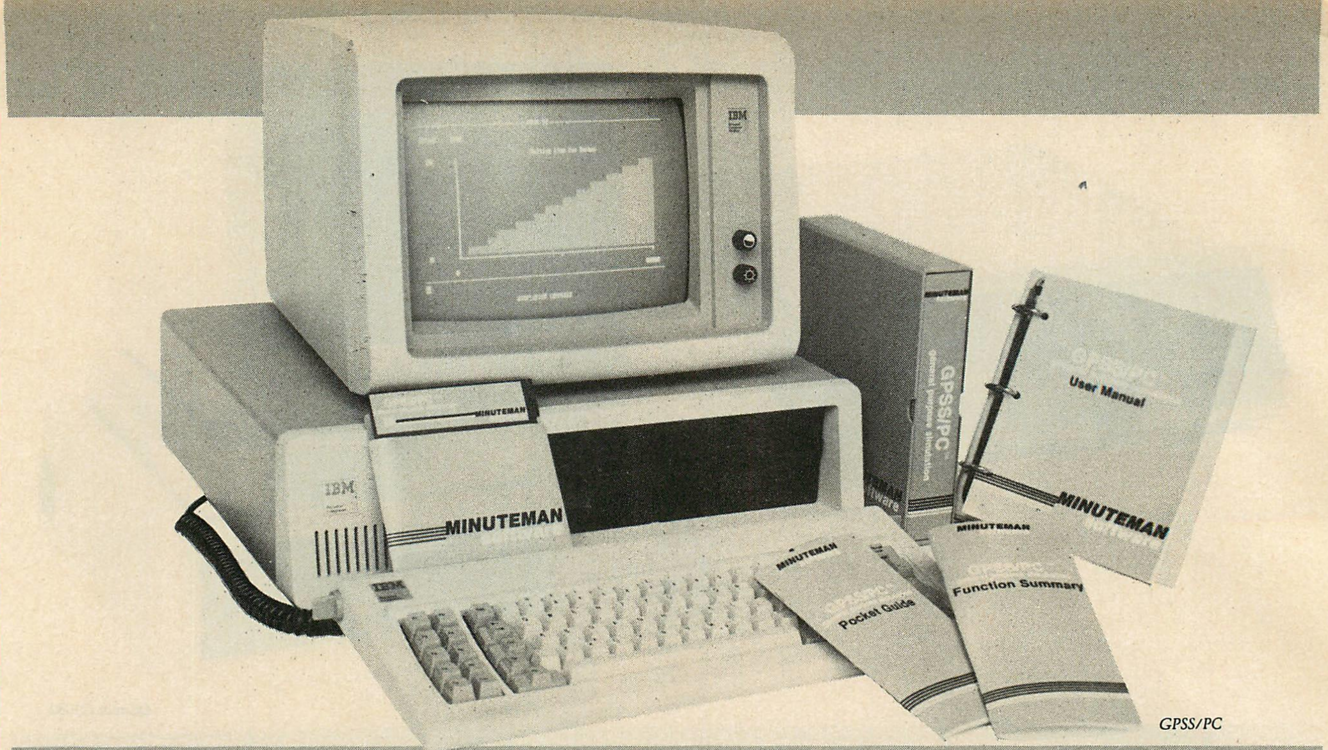
InfoCenter Software, Inc.
171 Main Street
New Paltz, NY 12561
914-255-8925

CIRCLE 474 ON READER SERVICE CARD



iLINK Screen

A new computer painting program from **Lifeboat Associates** claims to bring MacPaint-like graphics to the IBM PC. **Dr. Halo** is similar to Apple's MacPaint program in its use of icons and a mouse; Dr. Halo, however, also provides color. Because the program was written with the HALO Graphics Standard, it works with most of the popular graphics



GPSS/PC

boards sold for the IBM PC and compatibles. Microsoft, Mouse Systems, and Summagraphics mice are compatible with Dr. Halo, as are several digitizers. Images can be printed using the IBM and Epson FX and Graftrax printers and the IDS Prism color printer. $100.

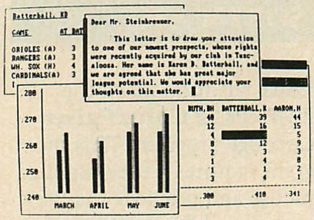
Lifeboat Associates
Department C
1651 Third Avenue
New York, NY 10128
212-860-0300

CIRCLE 472 ON READER SERVICE CARD

APX Core from **Application Executive Corporation** allows the user to run concurrently up to eight off-the-shelf IBM PC software programs. For example, text could be written on the screen while both a printer and a modem are in use. APX Core requires 48K of memory, one floppy disk drive, and no extra hardware. Data can be transferred from one program to another, and macros can be defined and saved on disk. $95.

Application Executive Corporation
600 Broadway
Suite 4C
New York, NY 10012
212-226-6347

CIRCLE 473 ON READER SERVICE CARD



Sample Screen for APX Core

Minuteman Software has introduced **GPSS/PC Version 1.0**, an efficient implementation of the popular simulation GPSS (General Purpose Simulation System). Using GPSS/PC, it is possible to predict the effects of managerial or engineering decisions on complex real-world systems. The package has more than 70 GPSS blocks and commands. Its features include "infinite" system-clock precision, "infinite" internal precision, full-period 32-bit random-number generators, and mathematic library functions. At least 256K memory is required; large simulations may require as much as 640K. $900 for one-time license fee.

Minuteman Software
P.O. Box 171
Stow, MA 01775
617-897-5662

CIRCLE 461 ON READER SERVICE CARD

Mylstar Electronics, Inc. has announced an enhanced symbolic debugger for the IBM PC. Called **MSD,** the product is intended to help programmers decrease the amount of time they spend debugging during the game-design process. MSD uses the same command structure as DEBUG and allows substitution of symbol names and mathematical expressions for hex values. It also contains commands for breakpoint editing, repeat loops, input/output redirection, on-line help, multi-command macros, and other time-saving features. The program requires an IBM PC with one disk drive, a minimum of 128K memory, and DOS 1.1, 2.0, or 2.1. $125.

Mylstar Electronics, Inc.
165 West Lake Street
Northlake, IL 60164
312-562-7400

CIRCLE 471 ON READER SERVICE CARD

REPORT CARD, a popular grading system published by **Sensible Software, Inc.,** has been revised for the IBM PC. The program allows teachers of all grade levels to record and monitor the progress of their students easily and quickly. The numerous printing and sorting options can be used to compute and post class rankings. Results can be displayed or printed by individual students or classes. $59.95.

Sensible Software, Inc.
24011 Seneca
Oak Park, MI 48237
313-977-3869

CIRCLE 470 ON READER SERVICE CARD

Version 2.0 of the **SuperSoft FORTRAN compiler** has been released by **SuperSoft, Inc.** It operates under MS-DOS and PC-DOS and contains large code and data space. The user can now have as many 64K code and 64K data segments as desired. In addition, version 2.0 follows the Intel calling format, so SuperSoft FORTRAN code can easily be linked with code from many other microcomputer languages. Debugging has been simplified by using in-line code. $425.

SuperSoft, Inc.
1713 S. Neil Street
P.O. Box 1628
Champaign, IL 61820
217-359-2112

CIRCLE 467 ON READER SERVICE CARD

Byte Gard

Keyswap

A powerful macro processor called **Keyswap** has been announced by **Rickerdata, Inc.** Keyswap runs in the background, is entirely transparent to the user, and is ready for macro definition at any time. The package has on-line help, an on-line listing of macros, and numerous keyswap command functions. It can be used with most popular applications software, including WordStar, Lotus 1-2-3, and dBASE II. $115.

Rickerdata, Inc.
P.O. Box 998
Melrose, MA 02176
617-662-0856

CIRCLE 460 ON READER SERVICE CARD

Two new products have been introduced by **Cincom Systems: PC CONTACT** and **SeriesOnePlus.** PC CONTACT is an integrated, interactive upload/download communications link between the IBM 370 architecture and the family of IBM PCs. It allows the user to extract corporate data off the mainframe host using MANTIS, Cincom's fourth-generation application-development language. The information can then be downloaded to a PC diskette.

SeriesOnePlus is an integrated decision-support software package for microcomputers. It includes word/text processing, spreadsheet, graphics, data organization, modeling, and report-writing software. $1,000 for the two packages together.

Cincom Systems, Inc.
2300 Montana Avenue
Cincinnati, OH 45211
513-662-2300

CIRCLE 469 ON READER SERVICE CARD

Intel Corporation has announced a new version (release 6) of **iRMX 86,** its real-time, multitasking, multiprogramming microcomputer operating system. This release extends the range of microprocessors supported by the operating system to include Intel's iAPX 188, 186, and 286, as well as the iAPX 88 and 86. The operating system can also support single-board computers and systems using these microprocessors. Release 6 of iRMX 86 will be used by OEMs as a system-management facility; it can be tailored to meet the demands of specific applications, which are typically in the machine-directed market—robotics, factory automation, process control, etc. OEMs can be working at the component, board, or system level. Initial license: $6,000.

Intel Corporation
5200 N.E. Elam Young Parkway
Hillsboro, OR 97123
503-640-7147

CIRCLE 468 ON READER SERVICE CARD

The **Hummingbird** operating system has been announced by **Computer Information Enterprises, Inc.** Written in PC native assembly language, Hummingbird is patterned after the IBM Series/1 operating system, EDX. It provides disk, timer, async, and multitasking support. Other facilities include: PC-Series/1 link (including data file and program transfers); support for task synchronization; timed wait; and PC-DOS file compatibility. $500.

Computer Information Enterprises, Inc.
2685 Dow Avenue, Suite B
Tustin, CA 92680
714-838-2171

CIRCLE 466 ON READER SERVICE CARD

PC/MPX, a nonpreemptive multiprocessing executive for the IBM PC, has been introduced by **The Austin Code Works**. Built for use with the DeSmet C compiler, PC/MPX allows a programmer or system designer to run multiple C functions concurrently. Functions share the CPU voluntarily by blocking to relinquish control to other members of the concurrent cooperative. $45.

The Austin Code Works
11100 Leafwood Lane
Austin, TX 78750
512-258-0785

CIRCLE 465 ON READER SERVICE CARD

OTHER WARES

Byte Gard head-cleaning disk is now available from **Precision Computer Products**. It is guaranteed for 260 cleanings or one year of use. The company recommends using its product for up to 30 seconds each working day to remove dirt, dust, and oxide deposits from disk read/write heads. $39.95.

Precision Computer Products
770 Welch Road
Suite 327
Palo Alto, CA 94304
800-321-2804

CIRCLE 457 ON READER SERVICE CARD

TECH BOOK
A Special Section for Product and Service Listings

Accessories/Supplies

RESET BUTTON FOR IBM PC
The button IBM forgot. A good reset button is a basic tool for PC (PC/XT) programmers. When your machine is hung up, a touch of the button will quickly reboot your computer. This kit installs easily and mounts out of the way behind your machine. Easy instructions included. Send $29.80 (add 6% in MN).
RESET CORPORATION
P.O. Box 14809
Minneapolis, Minnesota 55414
(612) 489-5892

IBM STYLE BINDERS, SLIPS
Complete program packaging line. D-ring cloth binders, slip cases, floppy pages, game portfolios. Continuous paper with three large holes, 20 lb. to go in binders. Blank disk envelopes. Function key cards tell user your F1-F10 meanings. Call, write for prices. Catalog. Fast service, low prices
ANTHROPOMORPHIC SYSTEMS LIMITED
376 E. St. Charles Road
Lomobard, IL 60148
(312) 629-5160

Computer/Insurance

INSURE YOUR COMPUTER
Our small computer policy covers your hardware, software & supplies against almost any physical loss or damage. Limited personal liability too. Business use is permitted. Available in most areas. Competitive rates. For more information & application contact.
COMPU SURANCE, INC.
PO Box 561952, Dept PCT
Miami, FL 33256-1952
(305) 665-6617

Computer/Services

INT'L DOCUMENTATION
INTERNATIONAL DOCUMENTATION provides reliable service in the preparation of foreign documentation & mktg. materials. Service includes translation, typesetting, & graphic art. Clear and accurate documentation in any language is a must. Let INTERNATIONAL DOCUMENTATION make your foreign documentation speak for you.
INTERNATIONAL DOCUMENTATION
120 Barranca Ave. Suite A
Santa Barbara, CA 93
109
(805) 965-4761/(213) 990-4886

Hardware/Add-on Boards

EPROM & MP PROGRAMMER
PC compatible APROM-2000 card can program 2716, 32, 32A, 64, 128, MCM 68764 EPROMS and also 8748/49/51 processors. The software (CP/M-86, MSDOS) can read, verify and program eproms and uses fast programming algorithm. The zero insertion socket is mounted on an external box. The external box for BI-POLAR proms and PAL is also available.
ADVANCED MICROCOMPUTER SYSTEMS, INC.
6802 N.W. 20th Ave.
Ft. Lauderdale, FL 33309
305-975-9515

256K NO SLOT MEMORY
IBM PC-1 owners. Let us install up to 256K RAM on your EXISTING SYSTEM BOARD using NO SLOTS. We improve your IBM PC so it can address the 64K RAM chip, 256K-$335 (expandable and warranted). DO-IT-YOURSELF KITS—PC-KPC-$69.95 ($99.95 assembled). 64K-RAMS-$5.85 ea.
ADD-MEM
22151 Redwood Rd.
Castro Valley, CA 94546
(415) 886-5443

MULTICOMM COMMUNICATIONS
ESE introduces two specially designed RS-232C serial communications adapters for the PC & XT and compatibles. MULTICOMM II has two ports, $225.00, and MULTICOMM IV, a four port unit, $445.00. Both have SES's selectable 64K I/O address decoders and use a single interrupt line regardless of the number of ports added.
ELECTRONIC SYSTEMS ENGINEERING CORP.
477 Congress Street, Suite 911
Portland, Maine 04101
MC/VISA. (207) 773-7778

FIXED DISK BIOS/BOOT
fiXT boots from DATAMAC, DAVONG, XEBEC, PERCOM, GREAT LAKES, ZOBEX, others. Adds XT-like BIOS interface for your disk to IBM PC or COMPAQ. Plug-in installation. DOS 2.0/3.2/reqd. Specify controller model with order $70 + $3 shpg. + tax. MC/VISA (optional volume support at additional cost.)
GOLDEN BOW SYSTEMS
P.O. Box 3039
San Diego, CA 92103
(619) 298-9349

DT2801 SERIES ANALOG I/O
Plug-in data acquisition boards with 8DI/16SE analog inputs, high or low level programmable gains, 2 analog outputs, 16 lines of digital I/O, DMA, on-board clock, and on-board microprocessor. Optional software subroutines and screw terminal panels.
DATA TRANSLATION INC.
100 Locke Drive
Marlboro, MA 01752
(617) 481-3700

Hardware Communications

TOTAL PC&XT COMMUNICATION
ESE introduces two extremely powerful communications adapters for the PC&XT & compatibles. One is a unique multi-interrupt RS-232 serial add-on for one OR MORE users; $139.50, and the second is a programmable 24 line parallel interface; $119.45.Both have ESE's full 64K selectable I/O address decoders.
ELECTRONIC SYSTEMS ENGINEERING
477 Congress Street, Suite 911
Portland, Maine 04101
MC/VISA (207) 773-7778

Hardware Disk Drives

8" & 5¼" FLOPPY DRIVES AND DISKETTES
• 8" flexible disk subsystem—Attractive thin-line design stacks above or below system's unit (19½" × 18" × 3½") ANT with complete documentation. Controller board and software available.
I-8480 DD, DS, 2.4 MB $1495
I-8481 SD, DS, 1.2 MB $995
• 5¼" internal disk drives—TANDON or Control Data—DS, 40 ATP1, 320 KB.
TM 100-2 $235 • CDC 9409 $249
5¼" half height internal disk drives available.
• Control Data diskettes—1240-00 5¼" SS/DD w/write protect notch in hub ring—bx of 10 $22. 1244-00 5¼" DS/DD $35. 1225-00 8" DS/DD wpn $39.50. VISA/MC.

MICROXPRESS

MICROXPRESS
305 S. State College, Suite 135
Anaheim, CA 92806
(714) 632-8512

Mailing Lists

IBM MAILING LISTS
Over 100,000 names of IBM personal computer owners (counts increase daily) available for rental on labels or magnetic tape. Total 600,000 including other brands. Plus IBM + compatible retail store + 50 other lists. Call or write for free catalog.
IRV BRECHNER
TARGETED MARKETING, INC.
Box 5125
Ridgewood, NJ 07451
(201) 445-7196

Security

DATA PADLOCK
DATA PADLOCK is a software implementation of the National Bureau of Standard Data Encryption Standard. Written in Assembler. DATA PADLOCK offers max speed with no increase in file size. An 8 character password provides the key for encryption. Protect sensitive data, text or program files from unauthorized access. Works with all DOS versions $150. VISA/MC.
GLENCO ENGINEERING
3920 Ridge Ave.
Arlington Hts., IL 60004
(312) 392-2492

Rates and Information

Listings are grouped by category and consist of a bold lead line (23 characters maximum), 7 lines of ad copy (45 characters per line), plus 4 lines of company name, address and telephone number.

Listings are available only on a 3 issue basis at $90. per issue ($270. total). Copy will have a set format and remain the same for all 3 months. Enhance the appearance of your ad by including your Logo and an additional cost of $25. per issue ($75. minimum extra charge). Pre-payment is required by check, money order, or American Express, Diners Club, MasterCard, Visa credit cards. Closing Date: 1st of 2nd month preceding cover date.

Send copy and remittance to PC TECH JOURNAL, TECH BOOK, 12th FLOOR, 1 PARK AVENUE, NEW YORK, NY 10016. Call (212) 503-4506 for additional information or assistance.

TECH BOOK

Software/Business

dINVOICER + PLUS

MENU-DRIVEN BILLING & ACCOUNTS RECEIVABLE MANAGEMENT SYSTEM. Produces Invoices, Aged Statements, Invoice Journals, Receipts Journals, Aged Customer Account Status, Past Due Notices, etc. Available with modifiable source code for dBASE II users $129 or as a stand-alone Run-time package $199.
DATAMAR SYSTEMS
819 Gage Dr.
San Diego, CA 92106
(619) 223-6444

EXPERT SYSTEMS

Create your own expert systems with EXSYS. All input is English text or menu selection. Uses full 1000K and probabilistic IF-THEN rules. Expert systems developed marketable without royalty. Demo disk including text of manual $10 (refundable). Full program—$200. Requires 128K or more.
EXSYS
P.O. Box 75158
Contr. Sta. 14
Albuquerque, MN 87194

PC ACCOUNTING SYSTEM

Professional Accounting for small business or home use. No codes to forget: all lookups are alphabetic. Shows effect of each entry on the screen in English. Supports 20 business units, 1000 account codes, and 10,000 customers or vendors. Reports by business or consolidated in detail/summarized on screen/printer. Color/mono dual/floppy/hard disk. $69.95 (VISA/MC/CHECK/COD/MO). NYS add 7% tax.
PEGASUS SYSTEMS, INC. Dept. T
30 Harbour Pointe
Buffalo, NY 14202
(716) 852-2969

Software/Business Opportunities

SOFTWARE AUTHORS

WE WILL CONNECT YOU TO MAJOR SOFTWARE PUBLISHERS FREE. We are in contact with many publishers looking for marketable software. If you have a quality software package or program and wish to have it published and distributed, contact:
SOFTSEARCH, Inc.
P.O. Box 281
Budd Lake, NJ 07828
(201) 627-1790

Software/Communications

TELEX LINK SOFTWARE

Link your computer to telex service via ITT, RCA or Western Union with Cawthon's telex software. Designed specifically for telex communications, Cawthon software is easy to use, has on-line help, self-test diagnostics, frequently called numbers, and an excellent User's Guide. Cawthon software is available for many computers and operating systems.

Cawthon Scientific Group

CAWTHON SCIENTIFIC GROUP
24224 Michigan Avenue
Dearborn, Michigan 48124
(313) 565-4000 Telex: 810-221-1265

Software/Compiler

DESMET C—$109

Full K&R C compiler, assembler, linker, librarian, full-screen editor and example software. Both 8087 and floating point libraries. OUTSTANDING PRICE/PERFORMANCE. Rated 1st or 2nd in August '83 BYTE benchmarks. No royalties on generated code. C Ware newsletter. Unlimited updates at $20 each. PC-DOS (Ver 1.1 & 2.0), generic MS-DOS and CP/M-86 support. $109 for complete package, shipping included. Now available with source level debugger. Price $159.

C WARE CORPORATION

C WARE CORPORATION—MAIL ORDER DEPT.
P.O. Box 710097
San Jose, CA 95171-0097

WIZARD C COMPILER

The power of C with the thorough diagnostics of PASCAL. Full LINT diagnostic integrated with compiler. Full UNIX SYSTEM 3 compatibility. Floating point library; fast compact code; 1 register variable. In-line assembly code. UNIX emulation library. Uses MICRO-SOFT linker. PC/MS DOS 2.0 only. $450.00.
WIZARD SYSTEMS SOFTWARE
11 Willow Court
Arlington, MA 02174
(617) 641-2379

C SOFTWARE DEVELOPMENT

- Full C Compiler per K&R
  Inline 8087 or Assembler Floating Point
  Full 1 Mb Addressing for Code or Data
- MS DOS 1.1/2.0 Library Support
  Program Chaining Using Exec
  Environment Available to Main
- c-window™ C Source Code Debugger

COMBINED PACKAGE $199
c-systems
P.O. Box 3253
Fullerton, CA 92634
(714) 637-5362

C-FUNCTION LIBRARY $29

100 C functions, macros, assembly functions. Screen, cursor, keyboard, string, BIOS and DOS access, of course. Plus BASIC equivalent, utility and many more you don't have but need! Just one can save you five times the cost of entire package. Manual w/source code $29.95. Manual + 2 disks: source + 5 libraries $49.95. Lattice or Microsoft C. (DeSmet and CI soon).
ENTELEKON
12118 Kimberley
HOUSTON, TX 77024
(713) 468-4412

SPARRY BASIC—B COMPILER

Powerful basic compiler that means business.
* Instructions support internal ISAM files.
* Four virtual screens, recall time <1 second.
* Use all use of all memory for basic programs.
* 13 significant digit integer math package.
* Demos with tutorial and reference manual. Price $159. Quantity discounts VISA/MC/CHECK. Mass. res. + 5% tax req. PC DOS with 128K.

SPARRY SOFTWARE LABS
PO Box 632 #A
Milford, MA 01757
(617) 473-5435

Software Development Tools.

THE FORMS DESIGNER™

Attention IBM Pascal, FORTRAN users! Save time in designing formatted screen I/O. Interactive Forms Editor allows you to draw lines and boxes, define fields, and edit text. Access forms or read keyboard entry by writing only one line of code. Provides sequential data retrieval and storage. Requires 128K RAM. Only $275 complete. Demo and manual $35. Call or write:
BIT SOFTWARE
PO Box 619
Milipitas, CA 95035
(408) 262-1054

FREE FREE FREE

BASIC AIDS FACT SHEETS and our guide titled "MAKE YOUR PC PROFITABLE" are sent free to persons who request them. Learn more about developing Structured Programs in BASIC. This new release of BASIC AIDS is the most powerful program DEVELOPMENT and DOCUMENTATION tool available!
TULSA COMPUTER CONSORTIUM
P.O. Box 707, 1004 North Dogwood
Owasso, OK 74055
(918) 747-0151

C-INDEX+

C index plus provides complete data management for C language applications development. Includes variable length data storage, B+ Tree ISAM indexing, interactive tutorial. Supported compilers. Lattice & CI-C86. No application royalty fee. Object code license: $400. Demonstration package: $25.
TRIO SYSTEMS
2210 Wilshire Blvd. Suite 289
Santa Monica, CA 90403
(213) 394-0796

SOFTWARE PERFORMANCE PROFILER

THE PROFILER™ will tell you where your program is spending it's time and allows you to do performance tuning on your code. Works with any language, data printed in histogram form. Easy to use and interpret. Complete user manual provided. Requires DOS 2.0 or better and 64K. $175.00 VISA/MC OK.
DWB ASSOCIATES
PO BOX 5777
BEAVERTON, OREGON 97006
(503) 629-9645

PASCAL/FORTRAN/C UTIL

Written in assembly language, Pascal/Fortran/C Utilities control screen, keyboard, graphics, music, lightpen, mouse, joystick, printer, RS—232, & I/O ports. Line/circle/ellipse/pie, clipping, windowing & animation. Terminal emulator, random #, BIOS/DOS calls. W/150 pg. manual, demos. Specify compiler. $119. Check/VS/MC.
SOFTWARE LABS
1221 Matisse Street
Sunnyvale, CA 94087
(408) 730-8108

RAPID BASIC PROGRAMMING

SAVE VALUABLE TIME WITH THESE EFFECTIVE TOOLS:
—R—Reference line numbers & variables used;
—A—Alter or find string in program or file;
—P—Peripheral input/output interchange prog;
—I—Include basic subroutine source modules;
—D—Duplicate variables in 2 modules listed.
SOURCE MICROSOFT BASIC FORMATS ALL FOR $49.95.
JAMES HALSTEAD & ASSOC.
1551 Plainfield Road
Joliet, IL 60435
(815) 725-0346

COBOL GENERATOR

Generates IBM PC or MS COBOL
- Screen Design/Data Dictionary
- Structured Program Generator
- Query/Report Generator

From $250—send for information
—The COBOL Experts—
- Manframe to PC conversions

BOMAC SYSTEMS
P.O. Box 10670
Pittsburgh, PA 15235
(412) 795-3581

TECH BOOK

TOOLS FOR PROGRAMMERS
• FILE TOOL KIT. COMPARE lists file differences, COMPRESS shrinks files, ENCRYPT secures valuable data, HEXDUMP lists files in hex and ASCII.
• PASCAL TOOLKIT. PXREF prints program cross-references; PNEAT prettyprints programs. PCASE highlights variables. Just $29.95 each, both for $49.95. VISA/MC accepted.
SEVEN VALLEYS SOFTWARE
P.O. Box 99
Glen Rock, PA 17327
(717) 235-5470

PADLOCK/PADLOCK II DISKS
PADLOCK furnishes the user with a method for providing protection against unauthorized duplication from DOS commands $99. PADLOCK II disks come preformatted with finger-print and serialization. PADLOCK II disks offer superior protection. Ask about our fast data encryption product. All work with hard disk, EXE/COM files and all DOS versions. MC/VISA.
GLENCO ENGINEERING
3920 Ridge Ave.
Arlington Hts., IL 60004
(312) 392-2492

FULL CONTROL FROM BASIC
Cut development time, add speed & power to your programs w/NLIB. A library of assembly language subroutines callable from BASIC: execute COM & EXE files, save & restore areas of the screen, clear & scroll windows, much more! Add DOS functions to your menus. Create fast help screens. Unlimited uses. All DOS Versions $49.95 + $2.50 s&h. MC/VISA/CHECK. PA + 6% tax.
NATIONAL SOFTWORKS
65 East Elizabeth Ave.
Bethlehem, PA 18018
(215) 867-4800

BASICWINDOW
Screen Generator and Processor for BASIC programmers. This package combines a Screen Painter for creating and modifying screens and Macro Language and Compiler for quicker coding and testing. Features COLOR, field variables, complete input editing, line drawing. For BASIC, BASICA, and BASIC Compiler users. Includes manual and reference card. $99.95 VISA/MC accepted.

BasicWindow™

G. FREEMAN & COMPANY, INC.
15 Albin Road
Stamford, CT 06902
(203) 327-9868

dSEARCHER QUERY UTILITY
A professional searching tool for dBASE II files. Uses simple query syntax with report, recall, and edit facilities for single file searches of any complexity. Creates reusable command programs to search unlimited number of files for "like" .data. Tutorial incl. $79.00. Visa/MC/COD. Call/write for info.
MICRO BYTE
P.O. Box 245
Burke, VA 22015
(703) 569-6473, (703) 451-8717

FIX DOS WITH A.K.A.!
Take full advantage of multiple directories—let unmodified programs access files outside of the current directory without pathnames. Alias file name to full paths, or direct searches to other directories according to name or extension, with multiple options. Operation is invisible—aliases are contained in the DOS 2.0 "environment." $60. MC/VISA.
SOFTSHELL TECHNOLOGYS
54 Walnut Hill Rd.
Newton, MA 02159
(617) 332-8186

BASIC RAM GRABBER II
With these 20 FAST subroutines, array sizes are limited only by amount of RAM installed on your PC. Chained programs can use up to 512KB for passing common variables. Software provided to reserve up to 512KB for string, integer, single and double precision arrays. Many other powerful features included. For both interpreted and compiled BASIC programs. $49.95 + $2 shipping. Accept check, MO, VISA/MC. All orders shipped within 5 days. Express mail add $10. For more information or to place an order, call or write NOW.
P&C Software
2152 37th Street
Los Alamos, NM 87544
(505) 662-7258

DOS PATH COMMAND FOR DATA
Now Lotus 1-2-3, dBASE II, WordStar and most others can access data files no matter where they are located! Works with hard disks and floppies. DPATH is a system stay-resident function that allows you to organize your disk any way you like. Includes a screen-oriented maintenance utility & a 60-page manual. DOS 2.0/compat. $25.
PERSONAL BUSINESS SOLUTIONS INC.
PO BOX 757
Frederick, MD 21701
(301) 865-3376

FILE DIFFERENCE PROGRAM
Compares two MS-DOS text files on a line by line basis. Produces a report of lines and blocks changed, inserted, and deleted. Extensive look-ahead and backtracking for intelligent operation on any programming language or text file. An invaluable programming tool, only $45. Send ck/mo.
DCP SOFTWARE
4125 Blackford Ave., Suite 205
San Jose, CA 95117

DISK MECHANIC
THE ULTIMATE Floppy Disk Backup & Repair Utility. Can back up ALL software protected disks written on the IBM PC. Works manually or automatically. Files or sectors can be restored, searched, examined & changed. Checks disk drive speed. Req. IBM PC, DOS 1.1 (128K)/2 (192K) + 64K if only 1 floppy drive. $73 ppd. USA
MLI MICROSYSTEMS
PO BOX 825, Dept. TB
Framingham, MA 01701 USA
(617) 926-2055 for info MC/VISA

EASY WINDOWS WITH C
Linked to your main C program, CRIOS provides easy screen management and yet is friendly to your end-user. Driven from a simple screen format string, it allows 8 windows, input and output variables, cursor positioning, scroll, change screen attribute, many other features. For Lattice C compiler and DOS 2.0. $49.95.
LRM SYSTEMS
4438 Donna Drive
Cleveland, Ohio 44143
(216) 383-9831

SDA FILEPAQ VERSION 1.0
Four DOS 2.n file utilities that can with a single command, on one or many directories; locate files and their file descriptions; generate sorted file lists by name, extension, size, and/or attributes; delete files; remove directories; modify file attributes. Wild cards used in path names. $29.50+$2.50 S/H
SDA Associates
PO Box 36152
San Jose, CA 95158
(408) 281-7747

SDA OKIPAQ VERSION 1.0
Three Okidate 92/93 printer utilities that allows automatic printing of all 255 IBM PC characters as well as medium resolution graphics screen images. In addition, the printer attributes can be easily set with a single command from the terminal. Extends the use of the Okidate 92/93. $29.50 + $2.50 Shipping/Handling.
SDA ASSOCIATES
P.O. Box 36152
San Jose, CA 95152
(408) 281-7747

FILEPATH VERSION 2.0
Unleash the power of DOS 2.0/2.1 subdirectories. Defines search paths for files like PATH does for commands. Wildcards used in path definitions. Eliminates the need for multiple Default, Help, Configuration, and Overlay files dBASE II, etc. $24.50 + $2.00 Ship/Handling.
SDA ASSOCIATES
P.O. Box 36152
San Jose, CA 95158
(408) 281-7747

SOFTWARE/WORD PROCESSING

PELADA TEXT ENGINEER
Word processing for scientific and multilingual text. All IBM characters. Super/subscripts, italics, underlining, bolding. With graphics hardware, you can use any symbols and any alphabet, up to 384 characters. All symbols visible on screen and printable. Easy English commands. $250.
PELADA INFORMATICS INC.
562 Johnson Street
Kingston, Canada K7L-2A1
(613) 549-1747

VOLKSWRITER DELUXE
Features TextMerge™ for customized mailing lists; multi-lingual, engineering & scientific symbols; horizontal scrolling to 250 characters; document size 1 megabyte; on-screen page endings, underline, boldface, double-strike, strike-through, multiple fonts & colors—fewest keystrokes per function. Only $295.
LIFETREE SOFTWARE INC.
411 Pacific Street
Monterey, CA 93940-2790

TECH BOOK

Software/ Languages

ADA ON A MICRO
JANUS/ADA is an exceptional programming language for software development. Being a subset of the Ada programming language JANUS/ADA implements those features of Ada which are directly needed by professional programmers. These features include separate compilation, user defined data types and helpful program debugging.
R.R. SOFTWARE
P.O. Box 1512
Madison WI 53701
(608) 244-6436

APPLICATION DEVELOPMENT
Build any application in a fraction of the time it would take with BASIC or COBOL. TPL combines conventional programming language constructs with a database manager, screen handler, and report generator in a powerful, yet flexible application development system. Full TPL system $250. Manual/Demo $50.
Randak Systems
346 N. Kanan Road, #204
Agoura, CA 91301
(818) 706-3737

Software/ Mailing Programs

PONY EXPRESS XL
Designed for commercial mailings or extra large lists. Super fast machine code multi sort. Carrier route sort. Prints labels: 1-5 across, single line format, or user-defined format. Also CHESHIRE FORMAT. Unlimited files. Multi file merge. Records can be selected by user-defined codes, specific zips or zip ranges. Phone list option. Unique correction & deletion routines for updating. Instant access to all records. Auto repeat for ease of record entry. Interfaces w/most word processors. $238.
COMPUTECH
975 Forest Ave.
Lakewood, NJ 08701
(201) 364-3005

Software/Operating Systems

MULTIPLE USERS UNDER PC DOS
MultiLink turns PC-DOS into an efficient multi-user multi-tasking operating system. Additional users are supported by attaching inexpensive CRT terminals or modems to serial ports on the PC, and can run normal applications designed for PC-DOS. Includes host communications software for public dial-in.
THE SOFTWARE LINK INCORPORATED
6700 23B Roswell Rd.
Atlanta, GA 30328
(404) 255-1254

Software/Sort

OPT-TECH SORT/MERGE
Extremely fast Sort/Merge program for the IBM-PC. Can sort or merge multiple files containing fixed or variable length records. Run as a DOS command or call as a subroutine, plus many other features. Now also sorts dBASE II files! Compare before you buy any other. Write or call for more info. $99.
OPT-TECH DATA PROCESSING
P.O. Box 2167
Humble, TX 77347
(713) 454-7428

INTERNAL SORT FOR BASIC
NSORT—A FAST & easy to use quicksort subroutine written in C & assembler, CALLable from BASIC. Sorts an unlimited number of records in memory on up to 8 keys of mixed type, ascending/descending. Like having a SORT built into the language. Works w/compatibles, Compiled and Interpreted, all DOS versions. $49.95 + $2.50 s&h. MC/VISA/CHECK. PA + 6% tax.
NATIONAL SOFTWORKS
65 East Elizabeth Avenue
Bethlehem, PA 18018
(215) 867-4800

Software/Terminal

BLUELYNX
Replace your dumb terminals with smart PCs and XTs. TECHLAND SYSTEMS is the nation's leading synchronous communications company with over 6,000 BLUELYNX' installed. The BLUE-LYNX hardware/software product emulates the 5251/12 for the S/34, S/36 and S/38. Now bundled with Autolynx for unattended operation ... only $745. BLUELYNX 3276 for the larger 43xx and 30xx mainframes, both SDLC and BSC versions, now comes bundled with Data Reader, a utility that facilitates the formatting of host files into Lotus 1-2-3, dBase, etc. ... only $795. Printer and up-and-download support available with both versions.

TECHLAND SYSTEMS INC.

TECHLAND SYSTEMS, INC.
25 Waterside Plaza
New York, NY 10010
(212) 684-7788

Software/ Typesetting

HIGH-TECH TYPESETTING
Transmit your text via toll-free lines directly to our fully automated typesetting system. $2. per K characters with a $5. minimum. Same day service. 200 typefaces in sizes up to 72 point. Send $15. + $3. shipping for our 220 page guidebook, or call toll free and use your MC, VISA or AMEX.
INTERGRAPHICS INC.
106A South Columbus Street
Alexandria, VA 22314
(800) 368-3342 or (703) 683-9414 in DC area.

Software/Utilities

IBM-PC COBOL PROGRAMMERS
Screen coding is fun, fast and accurate with Qwik-Screen Builder. "Paint" screen layout and create true COBOL source code for screen section in minutes. 100+ pages of documentation. Satisfaction guaranteed or money back. Needs DOS 1.1/2.0, two DS/DD drives, monochrome display and the COBRUN. EXE runtime module. $50/Visa/MC.
THE COMPUTER WORKS
P.O. Box 1371
Coeur d'Alene, Idaho 83814
(208) 667-6152

SNOOP DISKETTE UTILITY
Produce graphic maps of diskettes and files. Comprehensive diskette analysis. Add, delete or modify labels. Pageable directory. Change file attributes. Display, modify or search sectors or files. Much more, all in a single, easy-to-use program. Documentation included. Introductory price $40. SASE for information.
OTTER CREEK SOFTWARE
75 Hunting Lane
Goode, Virginia 24556

CONVERTS ANY CPM TO DOS
CROSSDATA CONVERTS ANY DATA/TEXT FILE format from CPMxx to MS/PC-DOS, CPMxx to CPMxx & MS/PC-DOS to CPMxx. Cross data runs on IBM PC or comparable computer using MS/DOS 2.0. Cross data is a self-contained program. It comes with over 24 mats and user can add own format. To order send $99. check or money order to:

AWARD SOFTWARE, INC.

AWARD SOFTWARE, INC.
236 North Santa Cruz Ave.
Los Gatos, CA 95030
(408) 395-2773

COPY PROTECTION BY MSD
The ULTIMATE diskette copy protection system designed by software developers and publishers. A variety of protection methods are available to suit your protection requirements. All IBM PC and XT environments are supported which includes. DOS, BASIC (compiled and interpretive). Pascal, P-System and stand-a-lone systems. Customized systems also available. Requires: 64K, two disk drives.
MICRO-SOFTWARE DEVELOPERS, INC.
214½ West Main Street
St. Charles, Illinois 60174
(312) 377-5151

Hard Disk DIRECT ACCESS™
The ULTIMATE utility for any hard disk user. DIRECT ACCESS is a powerful hard disk management tool. Organizes your software programs into a "user defined" menu system. Features attractive screen display, rapid access into an application using a single keystroke & easy to learn. Also handles batch files & complicated DOS commands. Order today. Only $44.50 + $2.50 shipping & handling. MC/VISA.
DELTA TECHNOLOGIES
P.O. Box 1104
Eau Claire, Wisconsin 54702
(715) 832-0958

TECH BOOK

C LIBRARY UTILITIES
Don't reinvent the wheel. Save time & money. Do a better job. Extensive library with source. Best screen handling available. Sound, graphics, DOS 2, printer, comm, date, time, and more. 150+ functions. DOS/BIOS gate gives complete control of your PC. C86, Microsoft, Lattice. Intro. offer $79. Demo $30.
ESSENTIAL SOFTWARE, INC.
PO BOX 1003
Maplewood, N.J. 07040
(914) 762-6605

BASIC DEVELOPMENT TOOLS
Increase your BASIC programming productivity with these preprocessors, filters, and utilities. All programs run from DOS level. All except NUMBER and LABEL work with binary format files! No need to save in ASCII.
* NUMBER programs written with your text editor using labels.
* LABEL programs written using line numbers. Identify entry points.
* CROSS REFERENCE variables, line references, and Basic reserved words.
* SMASH programs to smallest space. Removes remarks and extra spaces.
* COMPARE two Basic programs, documenting differences line-by-line.
* MERGE with delete. Combines two program fragments in line number order.
* LIST up to 20 programs with page headings and numbers without operator intervention.

Send $100 for complete kit, including manual and application notes, or write for info.
Datasmith, Inc.
Box 8036
Shawnee Mission, KS 66208
Phone: (913) 381-9118

CBTREE for C Programers
Provides enhanced file access function calls directly into C programs. Maintains balanced B-trees, supports unlimited number of data records, keys, data files & key lengths. Dynamic memory allocation. Fast, Flexible, efficient. Object modules avail. or major C compilers. No royalties; Source code separate. $179.
MICROCOMPUTING SERVICES
2009 Hileman Road
Falls Church, VA 22043
(703) 893-0118

800 ANSWERING SERVICE
Are you missing out on orders because your clients won't call collect or use their own nickel? Our 800 ANSWERING SERVICE will take your orders, inquiries and so forth. Economical, efficient. WORLD 800, 3435 St. Mary's Rd., Lafayete, CA 94549. (800) 235-6646 or (800) 235-6647 in CA.

TEXT FORMATTER ...
An assembly subroutine that when used in your BASIC and other programs formats text automatically on screen. NEVER splits line in middle of word! Justifies on screen! Centers. Clear and scroll sections. Total screen management. Save program development time. Distributed w/source code & several other programs. $19.95.
KARL THOMPSON
108-28 63rd Ave.
Forest Hills, NY 11375
(212) 896-4624

DEBUGGER/SOURCE INCLUDED
A debugger with FULL SCREEN format and SCREEN COEXISTENCE with test program (including graphics—even SCROLLS THROUGH LISTING FILES without disturbing the test session-all with NO COMMANDS and only 10 function keys! It's fast (assembler), small (60K), and only $75 with manual and includes the source. DOS 2.X IBM PC.
PC-TEST
for the software developer
P.O. Box 54068
San Jose, CA 95154

SOFTWARE/ ENGINEERING

PELADA TEXT ENGINEER
Engineering and Scientific Word processing. Technical symbols, special characters, super/subscripts, italics, underlining, bolding. With graphics hardware you can design any symbols you want, up to 384 different characters. All characters visible on screen and printable. Easy English commands. $250.
PELADA INFORMATICS INC.
562 Johnson Street
Kingston, Canada K7L-2A1
(613) 549-1747

ELECTRONIC CIRCUIT ANALYSIS
ECA is a high performance simulator for analog electronic circuits. Features include full editing, worst case, sensitivity analysis, ability to sweep components, 64 nodes. It is easy to use and very fast. Price $150. For all MS-DOS, CP/M, CP/M-86 computers.
TATUM LABS
P.O. Box 698
Sandy Hook, CT 06482
(203) 426-2184

HEAT TRANSFER ON THE PC
T/SNAP AND CPP1 are Thermal Engineering Programs. T/SNAP will solve steady-state and transient problems up to 250 nodes. Includes interactive input, FSE and plotting. CPP1 will design various size coldplates using compact fins. Calculates parametric electronic heat sink and pressure drop performance.
TECHNA/SOFT SYSTEMS
717 W. 14th Street
Long Beach, CA 90813
(213) 432-2704/823-6291

PC ENHANCEMENT HANDBOOK
THE IBM PC ENHANCEMENT HANDBOOK for SCIENTISTS & ENGINEERS. A How-To-Handbook with answers to all of your questions on PC enhancement. What can be done? How to do it? and How much will it cost? Includes Toll Free Hotline for application assistance & convenient one stop shopping. SPECIAL INTRODUCTORY OFFER $18.95.
CyberResearch, Inc.
P.O. Box 9565
New Haven, CT 06536
(800) 341-2525; (203) 436-2600 (CT)

SOFTWARE GENERAL

ATTENTION SOFTWARE AUTHORS
Our established literary agency is seeking to represent talented freelance programmers. We open doors for our clients at the leading software houses and negotiate the best deal possible. Put our years of experience to work for you. For further information on the benefits of representation contact:
THE ROBERT JACOB AGENCY
1642 Eveningside Drive, Suite 110
Thousand Oaks, CA 91362
(805) 492-3597

RENT PUBLIC DOMAIN SOFT/W
Rent our PC Disk Libraries for 7 days and copy them yourself. Hundreds of useful business, games and utilities FREE FOR THE TAKING!

IBM-PC SIG-135 Disks	$135.00
IBM-PC "Blue" 53 Disks	$50.00

Rental is for 7 days after receipt, 3 days grace to return. Most credit cards accepted.
NATIONAL PUBLIC DOMAIN
SOFTWARE CENTER
1533 Avohill Drive
Vista, CA 92083
(619) 941-0925. Info-(619) 727-1015. 24 hr. orders.

SOFTWARE/GRAPHICS

GEOGRAF™
GEOGRAF™ helps you prepare high-quality, customized graphs for business, engineering and science. GEOGRAPH™ cuts programming time by up to 80%. Callable from BASIC, FORTRAN or PASCAL. Versions available for most graphics cards and plotters. Fully compatible with Colcomp's FORTRAN standard plotting package.
GEOCOMP CORPORATION
342 Sudbury Road
Concord, MA 01742
(617) 369-8304

PRESS 'N' PLOT 2.0
Insert images into your text for correspondence that counts. Capture screen images from most graphics programs and merge them with text from most word processors. Can automatically wrap the text around the image. Supports most popular color or B-W dot matrix printers. $149.
AMERICAN PROGRAMMERS GUILD, LTD.
55 Mill Plain 17-5
Danbury, CT 06811
(800) 828-8088; in CT, (203) 794-0396

GRAPHICS LIB FOR TECMAR
TEK-MAR lets you do high-res graphics on your TECMAR Graphics Master. TEK-MAR is a library for use with MS Fortran. Features windowing, viewporting, clipping, axis rotation. Similar to Tektronix graphics. Includes screen dump/restore, Epson screen print, support for (Western Graphtec plotter. Requires MS-FORT 3.20, 320K, GMDEV.SYS (5.2). Price: $195.
ADVANCED SYSTEMS CONSULTANTS
18653 Ventura Boulevard, Suite 351
Tarzana, California 91356
(818) 990-4942

TEKTRONIX 4010 EMULATION
High resolution screen/printer graphics for the IBM PC. Full interactive capabilities with file transfer and cross-hair control. Utilizes PLOT 10, protocols. OFF-line review of graphics output. Supports IBM color/graphic or Hercules cards. Easy to use 4010 emulation at an affordable price. $80.00. Demo disk $5.
Technological Systems Group
5044 Haley Court
Lilburn, GA 30247
(404) 923-4980

TECH MART

FORTRAN or PASCAL PROGRAMMER?

READ THIS AD

GRAPHICS
- Text/graphics generics
- 2D interactive
- 2D plots (full support) including auto-scaling, auto-axis generation, auto-labeling, tabular/log/parametric curves, splines, bars, pies, you name it, we have it.
- 3D Plots incl. 2 hidden line removal options—not just old-fashioned wire frame.

PEN PLOTS
- Standard plotter primitives plus
- FULL 2D support plus
- Interface to screen graphics,
- Limited 3D.

Clear and complete documentation.

GRAFMATIC $135.
PLOTMATIC 135.

MICROCOMPATIBLES
11443 Oak Leaf Drive
Silver Spring, MD 20901
(301) 593-0683

CIRCLE 301 ON READER SERVICE CARD

ICs PROMPT DELIVERY!!!
SAME DAY SHIPPING (USUALLY)

64K DYNAMIC RAM ICs
200 ns $4.87, 150 ns $4.87

256K DYNAMIC RAMS
150 ns @ $35.77

Add 512K into XT or PPC @ $647!
With our proprietary information kit
Increase motherboard memory to 640 Kbyte!
CONSUME 0 EXPANSION SLOTS!

SPECIAL: 64K 150 ns @ $4.44
CONDITIONS: (1) Phone to confirm current price (2) Quantity: multiple of 24 or 25 (3) Prepaid, No Cancellations (4) Delivery 10 to 21 days (5) $10. shipping & handling (6) shipping insurance 30¢ per $100.

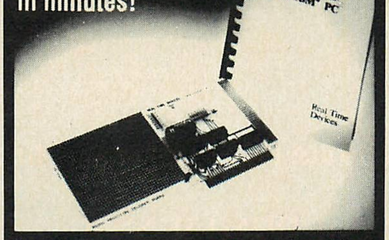
We Sell Other Memory ICs.
6264, 2764, 27128, 27256, etc.

Open 6½ days: we can Ship via Fed-Ex on Sat.

MasterCard/VISA or UPS CASH COD
Factory New, Prime Parts μP∞
MICROPROCESSORS UNLIMITED
24,000 South Peoria Ave.
BEGGS, OK. 74421 **(918) 267-4961**

Prices shown above are for July 30, 1984
Please call for new discount & current prices. Prices subject to change. Please expect higher prices on some parts due to world wide shortages. Shipping and Insurance extra. Cash discount prices shown. Small orders received by 6 PM CST can usually be delivered to you by the next morning, via Federal Express Standard Air @ $5.99!

Create your own unique PC interfaces in minutes!



The PD100 allows rapid development of specialized PC interfaces. It features a buffered data bus, switch-selectable address decoder, and a large prototype area (up to 40 IC sockets). The 116-page manual covers basic interfacing concepts and details implementing A/D, D/A converters, I/O ports, connection of transducers and dozens of useful circuits.

PD100 w/manual - $99 plus $3.50 P&H
Manual only - $20 Postpaid
PA residents add 6%
Check, MasterCard, or VISA

REAL TIME DEVICES
1930 Park Forest Ave.
P.O. Box 906
State College, PA 16804
(814) 234-8087

FREE DATABASE

**MANAGEMENT SYSTEM
PROGRAMMABLE RELATIONAL FULL
VERSION NOT A DEMO**

- Form Generator
- Subrecords
- Numeric Ranges
- Subtotals
- Report Generator
- Customizable Screens
- ASCII Output
- Tax Tables
- Menu Driven
- Creates Subfiles
- Sort on 3 Fields
- Recalculate Files
- 3 Files Open Simultaneously
- Global Field Changes
- Key strings to numbers
- Real time transfers between files
- Manual is on disk
- Written in Basic for modification
- 14 Ways to Calculate Fields
- Transfer Data between files
- No royalties on applications
- Retrieve on conditions for each field
- A "user supported" program
- Much More
- Over 15,000 Lines of Source Code Given
- Accounting, Payroll, inventory, A/R sample Applications
- for IBM PC with 128K
- PC Jr. with 128K & Cartridge Basic
- Compatibles with 192K & Microsoft Basic
- Mail Today for early delivery

POTOMAC PACIFIC ENGINEERING INC.
Box 2027, Gaithersburg, MD 20879 (301) 977-1489

- Copy the system from someone who already has one OR
- Send $10. to cover the cost of 2DSDD disks (or 2 flippy disks) 1st class mail, envelope, copying & handling.

Please send the data base to:
Name ____
Street ____
City, State, Zip ____

SALES POWER

PC: Tech Journal not only sells itself at a nice profit, it helps sell the hardware and software in your store too. Because it goes home with customers and continues to influence buying decisions long after you're closed for the day. Let's talk now.

**CALL COLLECT:
(212) 725-7679**
Or write:
ZD Ziff-Davis Publishing
One Park Avenue
New York, NY 10016

Minimum order, 10 copies. We pay all shipping costs.

SAVE

ON GROUP AND CORPORATE SUBSCRIPTIONS

If you're part of a user group or a group of people who use an IBM PC, PCXT, PCjr or compatible in the office or at school, you may be eligible for discounts to the world's leading IBM PC publications!
For more information, have a representative contact Dan Rosensweig at 212-503-5354

Ziff-Davis Publishing Company
One Park Avenue—4th Floor
New York, NY 10016

PC The Independent Guide To IBM Personal Computers

PC DISK New Programs on Diskette for the IBM PC, PCXT, PCjr and most Compatibles

TECH JOURNAL For Advanced IBM PC users.

PCjr. The Independent Guide To The IBM PCjr In Education, Home and Business

PC TECH JOURNAL PRODUCT INDEX

CIRCLE NO. 117 ON READER SERVICE CARD

INDEX TO ADVERTISERS

CALENDAR

OCTOBER

October 2-4
East Coast Computer Faire
Boston, MA
Sponsor: Computer Faire, Inc.
Contact: Dave Roth, 617-965-8350 (East Coast office); Nels Anderson, 415-364-4294 (West Coast office)

October 8-10
ACM 1984 Conference
San Francisco, CA
Contact: Karen A. Duncan, Health Information Systems, 15 Parsons Way, Los Altos, CA 94022, 415-948-3941

October 9-11
International Conference on Advances in Microprocessor Peripherals
Lausanne, Switzerland
Contact: Secretariat des Journées d'electronique et de microtechnique, Departement d'Electricité—EPFL, Chenin de Bellerive 16, CH-1007, Lausanne, Switzerland, 41-21-471111

October 15-17
4th Symposium on Reliability in Distributed Software and Database Systems
Silver Spring, MD
Sponsor: IEEE Computer Society, in cooperation with ACM
Contact: IEEE Computer Society, P.O. Box 639, Silver Spring, MD 20901, 301-589-8142

October 15-18
SOFTCON/FALL
Los Angeles, CA
Sponsor: Northeast Expositions
Contact: Northeast Expositions, 617-739-2000

October 16-18
UNIX EXPO
New York, NY
Sponsor: UNIGROUP of New York
Contact: National Expositions Co., Inc., 14 West 40th Street, New York, NY 10018, 212-391-9111

October 25-28
Philadelphia/Userfest
Philadelphia, PA
Sponsor: Northeast Expositions
Contact: Northeast Expositions, 617-739-2000

October 25-28
Twin Cities Computer Show and Software Exposition
Minneapolis, MN
Sponsor: CompuShows, Inc.
Contact: CompuShows, Inc., P.O. Box 3315, Annapolis, MD 21403, 301-269-7694

October 29-November 1
COMDEX/Europe/84
Amsterdam, The Netherlands
Contact: The Interface Group, 300 First Ave., Needham, MA 02194

NOVEMBER

November 1-4
Seventh Annual Northeast Computer Show
Boston, MA
Sponsor: Northeast Expositions
Contact: Northeast Expositions, 822 Boylston Street, Chestnut Hill, MA 02167, 617-739-2000

November 5-9
Eighth International Computer Software and Applications Conference (Compsac 84)
Chicago, IL
Sponsor: IEEE
Contact: Nick Marselos, AT&T Technology, 2600 Warrenville Rd., Lisle, IL 60532, 312-260-4104

November 12-15
IEEE International Conference on Computer-Aided Design
Santa Clara, CA
Sponsor: IEEE
Contact: William J. McCalla, Hewlett-Packard Co., 11000 Wolfe Rd., Cupertino, CA 95014, 408-257-7000, ext. 2225

November 30-December 2
Bits & Bytes, the First National Computer Experience for Kids
Anaheim, CA
Sponsor: Information Processing Group
Contact: Maryse Najar or Roger Fischer of Pollare/Fischer Communications, 213-478-0995

DECEMBER

December 6-8
1984 Real Time Systems Symposium
Austin, TX
Sponsor: IEEE
Contact: Miroslaw Malek, University of Texas at Austin, Computer Science Department, Austin, TX 78712, 512-471-5704

The Perfect Match

Davong combines good looks with hard disk and tape.

Davong has a beautiful way to solve both your hard disk storage and backup needs.

Our new DataSystem for IBM® PCs.

We offer a full range of disk sizes, all including a hard disk drive plus tape backup unit. And the capability of using the tape as a start/stop systems device.

In fact, this Davong DataSystem has a lot of other beautiful benefits, too. Like compressed or full-volume backup with file-by-file or full-volume restore. And automatic flaw mapping to ensure complete media interchangeability.

Our full XT compatibility actually offers performance a cut above other XTs. And if you want even higher performance and password protection, you can use our Multi-OS® software.

To top it off, there's the sleek, contemporary styling of our DataSystem. And a low price that's equally attractive.

Davong DataSystem for IBM PC's.

DAVONG

Davong Systems, Inc.
217 Humboldt Court
Sunnyvale, CA 94089
(408) 734-4900 Telex: 176386

®Multi-OS is a registered trademark of Davong Systems, Inc. ®IBM is a registered trademark of IBM Corp.

CIRCLE NO. 278 ON READER SERVICE CARD

The new HERCULES™ Color Card gives you the parallel port IBM® forgot–for $1 more.

And it fits into the XT's short slot.

The new Hercules Color Card gives you these five important features in a color graphics card.

1. A parallel printer port. If you buy IBM's color card, you won't get a parallel port for a printer. Ugh. You've just wasted a valuable slot for an extra printer card.

2. It's short. Our new Color Card uses advanced gate array technology to reduce the chip count by 30 and use less than half the board space IBM does. Which makes the new Hercules Color Card perfect for the short slot in an XT.

3. Fully IBM compatible. Any program that runs on IBM's color card will run on the new Hercules Color Card. With RGB or composite video, any IBM compatible monitor will work too.

4. Hercules Graphics Card compatible. If you have a Hercules Graphics Card (model GB101 or later), or plan to buy one, an ingenious software switch on our new Color Card allows both cards to be kept in the system at the same time.

5. Low price. At a suggested list price of $245, the new Hercules Color Card has all these features for only $1 more than IBM's color card.

Call for our free information kit. See why the company that made the first graphics card for the IBM PC still makes the best.

Hercules. We're strong on graphics.
800 255 5550 Ext. 423

CIRCLE NO. 151 ON READER SERVICE CARD

Address: Hercules, 2550 Ninth St, Berkeley, CA 94710 Ph: 415 540-6000 Telex: 754063 **Features:** 2 year warranty **Foreign distributors:** Compuserve/Canada; Reflex/U.K.; Computer 2000/W. Germany; Edisoft/France; SourceWare/Australia. Trademarks/Owners: Hercules/Hercules Computer Technology; IBM/International Business Machines.
This price comparison is based on the Suggested List Price as of 2.22.84 for the IBM Color Graphics Adapter.